

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Geração Procedimental baseada em Dados Afetivos

Artur Miguel Nóbrega Gomes



Mestrado Integrado em Engenharia Informática e Computação

Orientador: Doutor Rui Pedro Amaral Rodrigues

Co-orientador: Pedro Brandão Silva

Co-orientador: Pedro Nogueira

28 de Julho de 2014

Geração Procedimental baseada em Dados Afetivos

Artur Miguel Nóbrega Gomes

Mestrado Integrado em Engenharia Informática e Computação

Aprovado em provas públicas pelo Júri:

Presidente: Doutor António Augusto de Sousa

Arguente: Doutor Pedro Miguel do Vale Moreira

Vogal: Doutor Rui Pedro Amaral Rodrigues

28 de Julho de 2014

Resumo

A crescente evolução tecnológica e o aumento do volume de mercado da indústria dos videogames tem levado a uma constante melhoria da qualidade das produções. Esta traduz-se na capacidade que um jogo tem de fazer um jogador sentir-se imerso no mundo virtual. Para além da qualidade, os jogos têm ainda recebido um foco na expansão dos seus mundos virtuais, algo que sem as técnicas/métodos de geração procedimental de conteúdo teria custos muito elevados.

Em resposta à criação de experiências cada vez mais imersivas, nos últimos anos, os jogos têm vindo a ser alvo da aplicação de uma componente afetiva que tenta compreender melhor como melhorar a experiência para o público alvo. No entanto, a maior parte das soluções existentes falha numa das componentes mais essenciais: a transmissão de um conjunto de experiências e estados emocionais, imaginadas inicialmente pelos *designers*, para os jogadores. Com vista a resolver esta falha, nesta dissertação foi desenvolvida uma nova metodologia que tem como principal objetivo ajudar os *designers* a avaliarem os conteúdos gerados com base nas experiências e estados emocionais que pretendem transmitir aos jogadores.

As principais componentes desta metodologia são a *framework* onde ocorrem todas as ações relacionadas com o sistema de regras (desde a verificação à execução de regras), uma aplicação de recolha de dados afetivos (onde é efetuada a recolha dos dados dos sensores) e o jogo onde esta deverá ser aplicada. A metodologia usa estas três componentes para alterar o conteúdo do jogo em tempo real baseando-se nas regras definidas na *framework* e nos dados afetivos do jogador.

Por forma a provar que a metodologia funciona, foi feita a implementação no jogo EDuke32.

Para avaliar o caso de estudo foram usados 2 questionários e foram ainda recolhidos os dados obtidos pelos sensores durante os testes efetuados. Para os testes foram recrutados 13 utilizadores que jogaram a uma sessão repartida por quatro partes, uma por cada conjunto de regras: Boredom, Normal, Caos e Stress.

O primeiro questionário tinha como principal objetivo a identificação do conjunto de regras que os utilizadores jogaram. Os resultados mostram que para os utilizadores que testaram a aplicação a maioria identificou corretamente cada uma das partes da sessão (cerca de 62%).

O segundo questionário tinha como principal objetivo obter informações mais relacionadas com preferências e emoções sentidas durante o jogo. Em primeiro lugar, o conjunto de regras preferido pelos utilizadores foi o Caos (para cerca de 54% dos utilizadores).

As emoções que os utilizadores indicaram como sendo as mais significativas para cada conjunto de regras foram Relaxado para Boredom e Normal, Nervoso para Stress e Caos e Relaxado. A maior parte (46,2%) dos utilizadores acharam que o seu desempenho foi muito bom e o empenho foi bom. Através do teste de Friedman foi ainda possível demonstrar que relativamente ao grau de divertimento de cada uma das partes das sessões existem diferenças significativas.

Relativamente aos dados obtidos pelos sensores, devido a alguns problemas que ocorreram nos testes, não se obtiveram resultados conclusivos.

Com os resultados obtidos pelos questionários, é possível mostrar que a metodologia criada foi definida e implementada com sucesso.

Abstract

The growing evolution of technology and the growth of the video-games industry lead to a constant improvement on the quality of the video-games production. In the end the quality is measured by the ability of a game to make a player feel part of its virtual world. In response to the creation of more immerse experiences, in the latest years there have been efforts on developing an affective component that tries to create the best experience for the audience of a specific game. However, most of the existing solutions lack in one of the most important components. The transmission of a pack of experiences and emotional states, which have been imagined by the designers, to the players. With this problem in mind the goal of this master thesis is to create a new methodology that allows designers to evaluate the generated content based on the set of experiences and emotional states they want to transmit to players. Besides the quality, games have also been receiving focus on the expansion of their virtual worlds, which would have costs way to high if it were not for the procedural content generation techniques/methods.

The main components of this methodology are: the framework where all the rule management is done, the affective data collection application (where the sensor data is collected) and the game where it should be applied. The methodology uses this three components in order to alter the content of the game in real time, based on the rules defined on the framework and in the player affective data.

In order to prove the methodology works, an implementation of it was done for the game EDuke32.

To evaluate the case of study, 2 questionnaires were made and there there has also been a collection of the data received from the sensors during the tests. For the tests, 13 users were recruited. Each of the users played a session that was divided in 4 parts, one for each ruleset: Boredom, Normal, Chaos and Stress.

The first questionnaire main goal was to identify the set of rules that were played by the users during the session. The results show that for the 13 users that tested the application, most of them identified correctly all the 4 session parts (about 62%).

The second questionnaire main goal on the other hand was to obtain information related to the user preferences and to the emotions felt during the game session. That said, the favourite ruleset chosen was Chaos (for about 54% of the users).

The emotions that user pointed out as being the more significant for each of the rulesets were relaxed for Boredom and Normal, Nervous for Stress and Chaos. The major amount (46,2%) of the users though that their performance was good and that their effort was good. Using the Friedman test it was possible to demonstrate that the amount of fun felt between the four parts of the sessions was significantly different.

On the data obtained from the sensors, due to some problems that arose during the tests there were no conclusive results.

From the results obtained by the questionnaires it is possible to demonstrate that the created methodology was defined and implemented with success.

Agradecimentos

A elaboração deste trabalho não seria possível sem o apoio de todos aqueles que me ajudaram. Gostaria de agradecer ao professor Rui Rodrigues e aos co-orientadores Pedro Nogueira e Pedro Silva pela sua orientação, sem a qual não seria possível concluir este projeto.

Gostaria de agradecer também à minha família por todo o apoio que me prestou durante todo o trabalho. Aos meus pais por me motivarem a continuar a trabalhar e por serem uma fonte de inspiração, ao meu irmão por estar sempre presente e ao resto da família mais próxima pelas mensagens de apoio.

Os meus sinceros agradecimentos aos meus amigos por me ajudarem nos momentos mais difíceis e por estarem sempre presentes. Em especial os agradecimentos vão para: João Gonçalves, Telma Silva, João Lopes, José Bateira, Vitor Figueira, Sandra Gouveia, Ricardo Gouveia e Marlene Sá.

Finalmente a todos os meus outros amigos cujo nome não será mencionado por forma a que a lista não se torne muito extensa, mas que também contribuíram em muito para a conclusão deste trabalho, um muito obrigado.

Artur Gomes

“Sometimes I even amaze myself.”

Duke Nukem em Dukem Nukem 3D

Conteúdo

1	Introdução	1
1.1	Motivação e Objetivos	2
1.2	Estrutura da Dissertação	2
2	Revisão Bibliográfica	3
2.1	Teorias da emoção	3
2.2	Sensores	8
2.3	Reconhecimento Emocional	10
2.4	Geração Procedimental	12
2.4.1	Características	12
2.4.2	Técnicas	14
2.4.3	<i>Frameworks</i>	17
2.5	Geração Procedimental Afetiva	19
2.6	Outros Trabalhos Relevantes	22
2.7	Resumo	23
3	Metodologia	25
3.1	Arquitetura	25
3.2	Componentes	26
3.3	Regras	28
3.4	Aplicações	29
3.5	Resumo	29
4	Implementação	31
4.1	Caso de Estudo	31
4.2	A Solução	32
4.2.1	Arquitetura	32
4.2.2	Protocolo de Comunicação	33
4.2.3	Sistema de Regras	34
4.3	Regras	36
4.4	Resumo	37
5	Resultados	39
5.1	<i>Setup</i> Experimental	39
5.2	Validação e Avaliação	40
5.3	Análise dos resultados e Discussão	43
5.4	Conclusões	48

CONTEÚDO

6	Conclusões e Trabalho Futuro	49
6.1	Satisfação dos Objetivos	49
6.2	Trabalho Futuro	50
	Referências	53
A	Anexos	59
	Índice Remissivo	65

Lista de Figuras

2.1	Representação da estrutura OCC	4
2.2	Roda das Emoções de Plutchik	5
2.3	Representação do espaço AV	6
2.4	Regiões do Cérebro	7
2.5	Cubo das emoções de Lövheim	7
2.6	Sensores de EMG e músculos faciais	8
2.7	Sensores ECG e intervalo RR	9
2.8	Múltiplos sensores	10
2.9	Deslocamento do ponto médio em duas dimensões	14
2.10	Representação de três iterações do algoritmo Game of Life	17
2.11	Vista geral da arquitetura dos princípios da adaptabilidade de jogos	18
2.12	Geração procedimental baseada em ações	19
3.1	Arquitetura da solução proposta	26
4.1	Botão de fim de nível (Duke Nukem 3D)	32
4.2	Arquitetura da solução	32
4.3	Protocolo de Comunicação	33
4.4	Editor XML: Declaração da ação luz ambiente	35
4.5	Editor de condições e ações	35
4.6	Editor de regras	36
5.1	Protocolo Experimental	41
5.2	Ordem de preferência por partes de sessão	44
5.3	Empenho e Desempenho	46
5.4	Atividade e valência para conjunto de regras Boredom	47
5.5	Atividade e valência para conjunto de regras Stress	47
5.6	Atividade e valência para conjunto de regras Normal	47
5.7	Atividade e valência para conjunto de regras Caos	48
A.1	Atividade e valência para o utilizador 1	60
A.2	Atividade e valência para o utilizador 2	60
A.3	Atividade e valência para o utilizador 3	60
A.4	Atividade e valência para o utilizador 4	60
A.5	Atividade e valência para o utilizador 5	61
A.6	Atividade e valência para o utilizador 6	61
A.7	Atividade e valência para o utilizador 7	61
A.8	Atividade e valência para o utilizador 8	61
A.9	Atividade e valência para o utilizador 9	62

LISTA DE FIGURAS

A.10 Atividade e valência para o utilizador 10	62
A.11 Atividade e valência para o utilizador 11	62
A.12 Atividade e valência para o utilizador 12	62
A.13 Atividade e valência para o utilizador 13	63

Lista de Tabelas

5.1	Identificação de cada uma das partes da sessão	43
5.2	Percentagem de keywords por conjunto de regras	44
5.3	Resultados do teste de Friedman	46

LISTA DE TABELAS

Abreviaturas

ANN	Redes Neurais Associativas (do inglês: <i>Associative Neural Network</i>)
AV	Atividade-Valência
BVP	Pressão arterial (do inglês: <i>Blood Variation Pulse</i>)
DDN	<i>Dynamic Decision Network</i>
DFA	<i>Discriminant Function Analysis</i>
DT	Árvores de decisão (do inglês: <i>Decision Trees</i>)
ECG	Eletrocardiograma
EEG	Eletroencefalografia
EMG	Eletromiografia
FCM	<i>Fuzzy C-Means</i>
FKM	<i>Fuzzy K-Means</i>
GSR	<i>Galvanic Skin Response</i>
HR	Ritmo Cardíaco (do inglês: <i>Heart-Rate</i>)
IAPS	<i>International Affective Picture System</i>
KNN	<i>K-Nearest-Neighbor</i>
NN	Redes Neurais (do inglês: <i>Neural Networks</i>)
PCG	Geração Procedimental de Conteúdo (do inglês: <i>Procedural Content Generation</i>)
RF	<i>Random forests</i>
SC	Condutância da Pele (do inglês: <i>Skin Conductance</i>)
SVM	<i>Support Vector Machine</i>

Capítulo 1

Introdução

O desenvolvimento de jogos complexos é uma tarefa que consome bastantes recursos e tempo. Com o desenvolvimento de novas tecnologias e com o aumento do poder de processamento dos computadores, a complexidade de jogos também tem aumentado, o que implica um aumento significativo na quantidade e complexidade de conteúdo que tem que ser criado. Este aumento de criação de conteúdo traz também um aumento dos custos associados ao desenvolvimento dos jogos, tanto a nível monetário como a nível do tempo necessário para que se conclua o projeto.

Por forma a reduzir estes custos e a aumentar a complexidade e detalhe dos conteúdos têm sido desenvolvidas ferramentas de geração procedimental (algoritmos que criam conteúdo de forma automática ou semi-automática). Estas ferramentas de geração procedimental podem ser usadas para criar um mundo virtual completo, ou para criar um mundo virtual, que será depois usado como ponto de partida pelos *designers* do projeto, que deverão depois alterar o conteúdo, de modo a que este seja adequado ao contexto do jogo em desenvolvimento [CBPD11]. Se os mundos virtuais criados forem suficientemente variáveis e em tempo real, existe a possibilidade de gerar mundos virtuais infinitos. Assim, é possível criar novos tipos de experiências, que permitem aos jogadores explorar o mundo sem que este se torne repetitivo e entediante [TYSB11].

Para que um jogo seja bem sucedido não é apenas necessário criar mais conteúdo ou simplesmente aumentar a sua complexidade, é também crítico que exista uma preocupação com a experiência afetiva do jogador [JW03]. É portanto reconhecido por *designers* e investigadores que o aumento da experiência afetiva do jogador faz com que um jogo seja mais agradável do que um jogo que pouco ou nada evoca emoções num jogador [Haz06].

Os avanços tecnológicos na área do reconhecimento de emoções têm vindo a permitir que tal como nos livros e filmes, também nos jogos comece a existir uma transmissão de emoções, o que resulta em jogos mais envolventes a nível emocional. Isto implica portanto, que esta transmissão de emoções entre os jogos e os jogadores, está cada vez menos limitada pelas tecnologias existentes e mais dependente da imaginação dos *designers* [GDA05].

1.1 Motivação e Objetivos

Embora as ferramentas usadas para criar jogos tenham evoluído bastante ao longo dos últimos anos, existe uma falha ao conseguir captar e implementar as intenções dos *designers* [TSBdK09]. Esta falha faz com que os *designers* não consigam transmitir corretamente ao jogador o conjunto de experiências e estados emocionais específicos que haviam sido imaginados inicialmente.

Por forma a colmatar este problema esta dissertação tem como principal objetivo a criação de uma ferramenta que venha facilitar a modelação de níveis com base nas intenções dos *designers*. Para tal será necessário, mais especificamente atingir os seguintes objetivos:

- Investigação dos métodos que mais se adequam à geração procedimental afetiva;
- Criação de uma interface que mapeie os dados dos sensores em valência e atividade;
- Criação de uma especificação que permita mapear as emoções a funções de geração de níveis, permitindo a rápida e fácil alteração do conteúdo gerado.
- Criação de uma nova metodologia que permita através do estado do jogo e dos dados afetivos de um jogador, alterar o conteúdo gerado num jogo.
- Desenvolvimento de uma prova de conceito com um jogo, para validar a metodologia criada

1.2 Estrutura da Dissertação

Para além da introdução, esta dissertação contém mais 5 Capítulos. No Capítulo 2 é descrito o estado da arte e são apresentados trabalhos relacionados. No Capítulo 3 é efetuada uma descrição da Metodologia criada e são apresentadas algumas instruções para a implementação da mesma. No Capítulo 4 é apresentada a solução que consistiu na implementação da metodologia no jogo EDuke32. No Capítulo 5 são apresentados os resultados dos testes efetuados com os utilizadores recrutados. Finalmente no Capítulo 6 são apresentadas as conclusões sobre os resultados obtidos.

Capítulo 2

Revisão Bibliográfica

Nos últimos tempos têm surgido várias soluções para identificar as emoções de um utilizador ao utilizar determinadas aplicações, assim como várias soluções na área de geração procedimental.

As técnicas de geração procedimental baseiam-se em algoritmos que criam conteúdo (semi-) automaticamente através de regras. Embora os métodos de geração procedimental sejam por natureza algo específico de cada género, estilo e ambiente de jogo [Hen12], tem havido uma evolução a nível gramatical, que é específico para determinados tipos de modelação 3D. Ainda assim, é possível notar que tem existido também alguma preocupação na criação de soluções que permitam facilitar a modelação de uma forma mais genérica.

Os primeiros três subcapítulos deste capítulo dizem respeito a uma análise mais orientada à parte da deteção de emoções enquanto que o conteúdo dos últimos subcapítulos está mais orientado à geração procedimental. No subcapítulo 2.1 são discutidos os espaços emocionais e alguns modelos da teoria da emoção. No subcapítulo 2.2 são demonstrados alguns dos sensores mais utilizados para a obtenção de medidas fisiológicas. São apresentados alguns dos métodos, algoritmos e resultados obtidos dos trabalhos estudados ao longo do subcapítulo 2.3. No subcapítulo 2.4 é feita a apresentação de algumas das características dos métodos de geração procedimental, assim como de algumas *frameworks*. No subcapítulo 2.5 são apresentados os trabalhos estudados na área da geração procedimental afetiva. Alguns trabalhos relacionados ou com alguma semelhança ao desenvolvido neste projeto são apresentados no subcapítulo 2.6. Finalmente no subcapítulo 2.7 é feito um resumo dos trabalhos estudados.

2.1 Teorias da emoção

O uso de emoções em sistemas de software tem sido alvo de grande interesse nos últimos anos, como meio de melhorar a interação pessoa-computador. Com vista a melhorar este processo de interação muitos autores têm vindo a realizar investigação na área da teoria das emoções. No trabalho de Robert Plutchik [Plu91] são levantadas algumas questões cuja teoria das emoções deve

conseguir responder, tais como: O que é uma emoção? O que provoca emoções? Qual a função das emoções?

Relativamente ao trabalho de Ortony, Clore e Collins [OCC90] foi apresentado o modelo OCC - cujo nome vem das iniciais dos seus nomes - que assume a existência de três aspetos ou alterações no mundo nas quais um indivíduo se pode focar, nomeadamente, eventos, agentes e objetos. Segundo os autores um indivíduo foca-se num evento porque está interessado nas suas consequências, em agentes devido às suas ações e em objetos por causa do interesse em certos aspetos ou propriedades dos mesmos. Este modelo tem por base a ideia de que as emoções partem de reações baseadas na valência (positiva ou negativa) e que qualquer destas reações é uma perspetiva sobre o mundo.

As razões pelas quais os autores distinguem os agentes, objetos e eventos são: em primeiro lugar, os eventos são simplesmente interpretações que as pessoas fazem de acontecimentos e são considerados independentes de quaisquer crenças que estas possam ter sobre as causas possíveis ou reais. Segundo, os objetos são vistos não como objetos em si, mas antes como aquilo que aparentam ser. Finalmente os agentes, que são distinguidos pelo facto de serem causas ou contribuidores para certos eventos.

Feita esta distinção, os autores propõem uma estrutura que demonstra como através destas três formas de reação a ações do mundo, se podem extrair reações afetivas (figura 2.1).

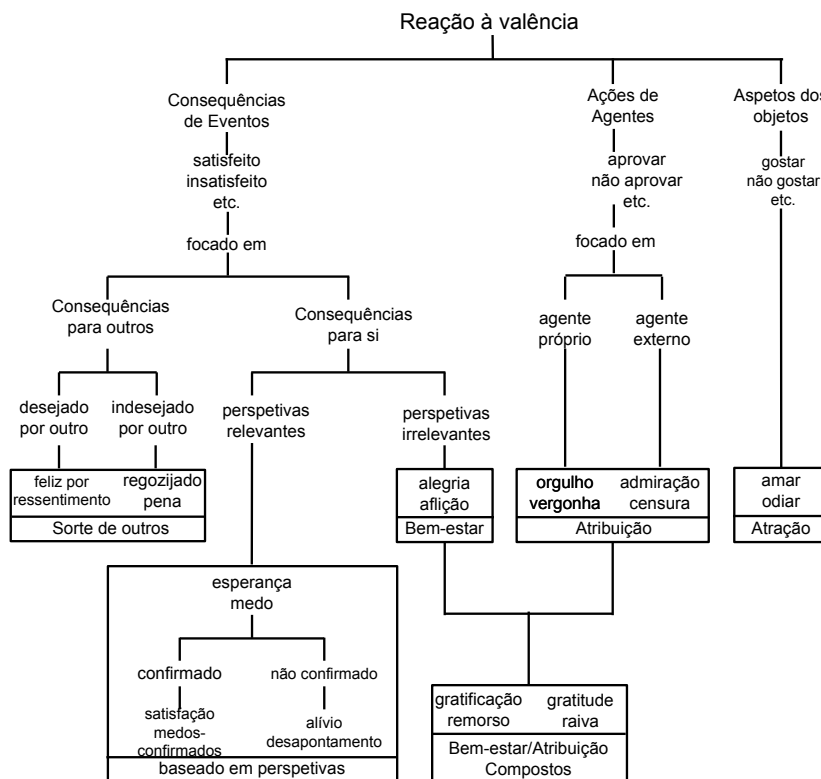


Figura 2.1: Representação da estrutura OCC (Adaptado de [OCC90])

Em 1991 Robert Plutchik [Plu91] apresenta o modelo denominado de Roda das Emoções. Neste modelo o autor indica a existência de oito emoções básicas que dão origem a várias outras emoções através da combinação e alteração de intensidade destas.

Um esquema do modelo pode ser observado na Figura 2.2. No centro estão representadas as oito emoções básicas. Entre a margem e o centro as emoções variam a nível de intensidade e entre duas emoções ocorrem emoções combinadas, como por exemplo *alegria + antecipação = otimismo*.

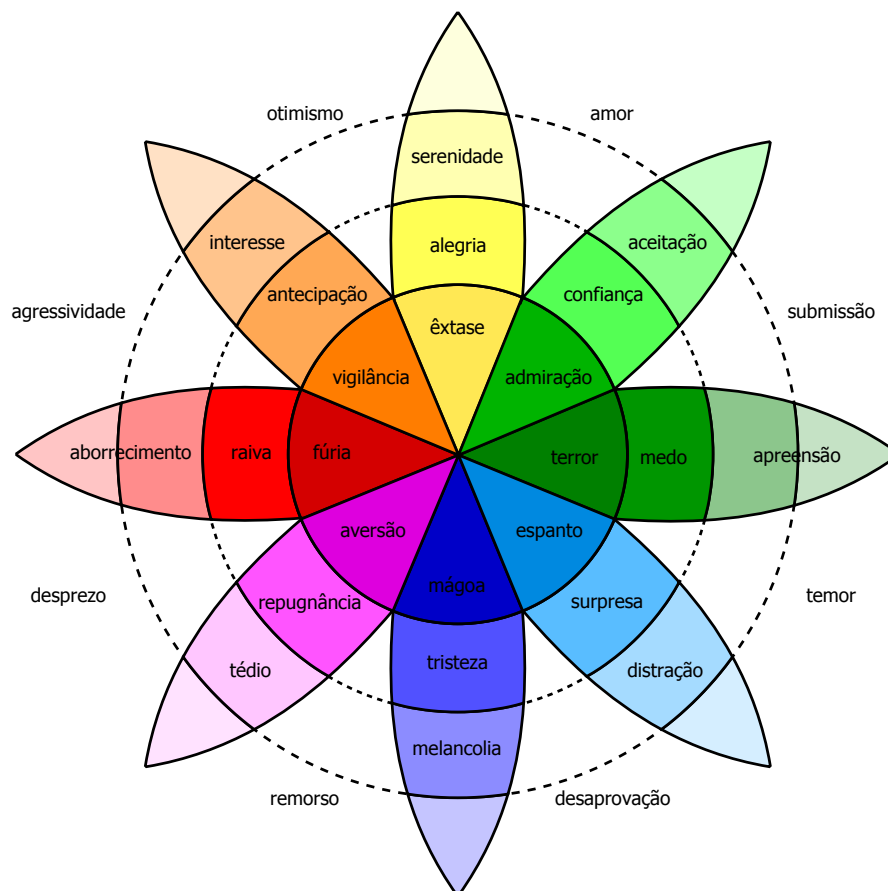


Figura 2.2: Roda das Emoções de Plutchik ¹

O modelo que foi usado na maior parte da literatura estudada para esta dissertação baseia-se na atividade e na valência por forma a classificar as emoções. Um exemplo da classificação de emoções no espaço AV (*arousal-valence*) é apresentado na figura 2.3.

A valência e a atividade variam de acordo com as descrições abaixo:

¹Fonte: http://commons.wikimedia.org/wiki/File:Plutchik-wheel_de.svg

Valência: mede se um estado afetivo é positivo ou negativo. Para valores muito altos indica que o sentimento é agradável, por outro lado, para valores muito baixos indica que sentimento é desagradável.

Atividade: indica a atividade mental ou física de um indivíduo. Para valores muito baixos indica sonolência, enquanto que para valores muito altos indica que o indivíduo está mais ativo.

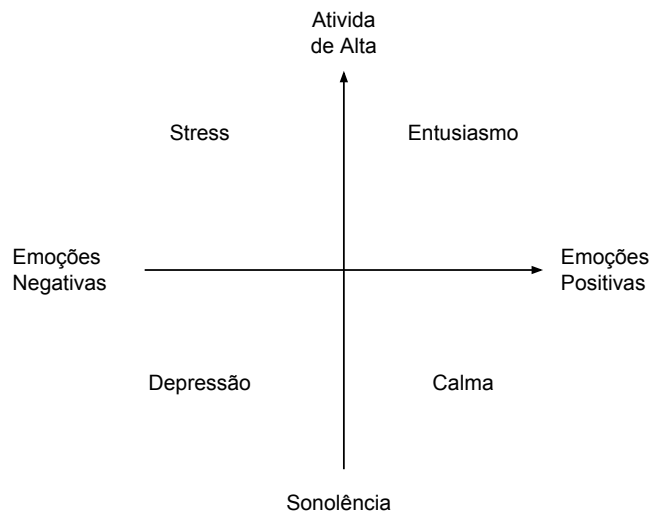


Figura 2.3: Representação do espaço AV (Adaptado de [MA07])

A ausência de escala na figura 2.3 deve-se ao facto da inexistência de *guidelines* para transformar o sinal de entrada dos sensores para no espaço de valência e atividade [MA07].

No artigo de Louis Schmidt e Laurel Trainor [ST01] são descritos três modelos que relacionam uma ou ambas as componentes do espaço AV com a atividade cerebral.

No primeiro sugerido por Davidson, Schwartz, Saron, Bennett e Goleman em 1979, foi discutido que as emoções são organizadas à volta de tendências do tipo aproximação-prevenção e diferenciadas na região frontal do cérebro (Figura 2.4). Na região esquerda frontal é possível identificar experiências positivas como alegria, interesse e felicidade. Na região direita frontal é possível identificar experiências negativas como medo, repugnância e tristeza. O segundo modelo descreve o papel da ativação frontal absoluta do cérebro na intensidade da emoção. O terceiro e último modelo considera ambas as dimensões. Neste modelo a valência é modulada da mesma maneira que no primeiro modelo de Davidson e outros, no entanto, neste caso a atividade é associada à ativação da região parieto-temporal do cérebro (Figura 2.4).

Mais recentemente Hugo Lövhelm [L12] cria um modelo tridimensional que visa representar as emoções com base em três monoaminas (substâncias bioquímicas que atuam no corpo humano), norepinefrina, a serotonina e a dopamina, que são essenciais no controlo de comportamentos e

² Fonte: http://upload.wikimedia.org/wikipedia/commons/0/0e/Lobes_of_the_brain_NL.svg

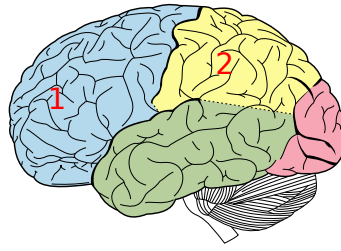


Figura 2.4: Regiões do Cérebro²; 1) Região Frontal; 2) Região Parietal

emoções. O mapeamento das emoções no espaço tridimensional é feito através de um cubo (Figura 2.5). Os três eixos ortogonais correspondem às três monoaminas.

Em cada vértice do cubo está mapeada uma certa emoção. Estas oito emoções foram descritas por Tomkins em trabalhos anteriores e podem ser decompostas em três grupos. Emoções positivas, sendo estas interesse/entusiasmo e satisfação/alegria. Emoções neutras surpresa/sobressalto. As restantes cinco emoções pertencem ao grupo das emoções negativas e são respetivamente, aflição/angústia, medo/terror, vergonha/humilhação, desdém/repugnância e raiva/fúria. Cada uma das emoções é representada por duas palavras, o que possibilita a representação da mesma emoção com intensidades diferentes. Em cada um dos vértices do cubo está representada uma emoção no seu estado mais extremo e algures a meio das linhas entre os vértices e o centro do cubo, estão representadas as emoções com intensidades inferiores, que representam um estado mais ou menos próximo de um estado neutro.

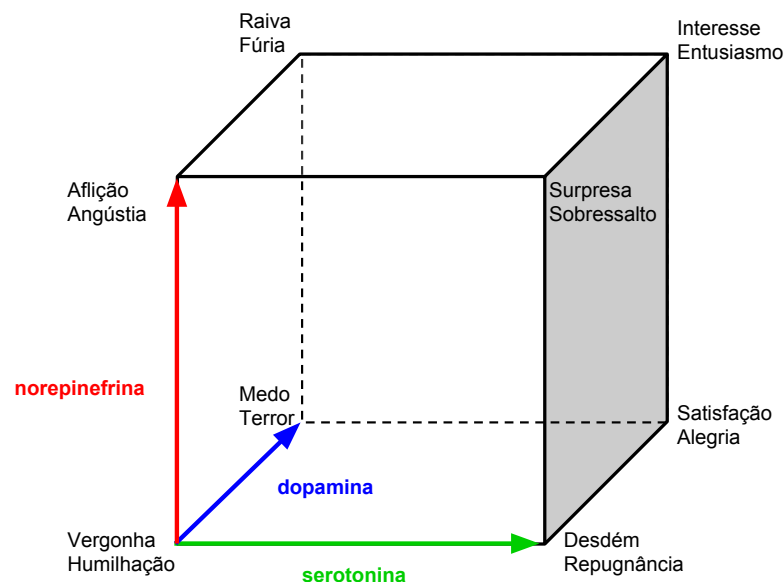


Figura 2.5: Cubo das emoções de Lövheim (Adaptado de: [L12])

No trabalho de José Serra e Pedro Nogueira [SN13] foi apresentado um novo método onde são simuladas iterações semelhantes às humanas através de agentes.

Este método tem como principal objetivo simular as alterações que ocorrem na componente emocional do comportamento humano quando ocorre um estímulo externo.

A arquitetura deste método é composta por três camadas. A primeira que é a da abstração de dados de entrada que é responsável por mapear os estímulos aos seus pesos emocionais. A segunda é a camada de avaliação dos estímulos que se mantém a par das emoções dos agentes e que altera de acordo com o estímulo e personalidade observados. A última camada é a camada de abstração de saída que está encarregue do processo de tomada de decisão.

2.2 Sensores

Ao longo dos anos têm-se usado várias técnicas para tentar obter as emoções dos utilizadores, entre as quais se podem destacar os sinais fisiológicos, expressões faciais, estudo de comportamentos e de padrões, entre outros. No entanto, são os sinais fisiológicos que têm obtido melhores resultados, pelo que neste subcapítulo serão apresentadas algumas das medidas fisiológicas e os sensores usados para as obter.

Eletromiografia (EMG) refere-se à atividade de um músculo ou frequência da tensão muscular de um certo músculo. O nível absoluto da tensão de um músculo é dependente do músculo escolhido para recolher a medida [HGSW04]. Um exemplo deste tipo de sensores pode ser encontrado nos trabalhos de Richard Hazlett e Regan Mandryk et al. [Haz06, MA07], em que foi usada a EMG facial. No trabalho de Richard Hazlett [Haz06], o autor concluiu que, através do músculo zigomático (Figura 2.6), se podem detetar reações positivas e através do músculo corrugador (Figura 2.6), se podem identificar reações negativas de um indivíduo. Na Figura 2.6 à esquerda está ainda representado um exemplo de um sensor deste tipo.

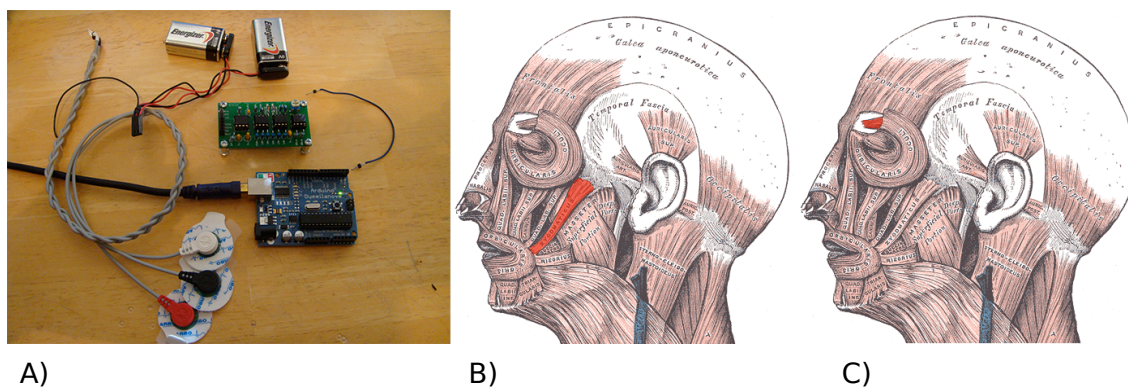


Figura 2.6: A) Exemplo de um sensor de EMG; B) A vermelho está representado o músculo zigomático; C) A vermelho está representado o músculo corrugador ³

³Fontes:

A) <http://www.instructables.com/id/Muscle-EMG-Sensor-for-a-Microcontroller/>

B) <http://upload.wikimedia.org/wikipedia/commons/7/7e/Zygomaticus.png/>

C) http://upload.wikimedia.org/wikipedia/commons/b/b4/Corrugator_supercilii.png/

Resposta galvânica da pele GSR⁴ é facilmente medida como resistividade ou condutividade da pele (SC⁵). As alterações são medidas através de um tipo específico de glândulas sudoríparas chamadas glândulas écrinas. Estas glândulas estão localizadas nas palmas das mãos e nas solas dos pés e respondem não só a alterações de temperatura corporal mas também às fisiológicas. As glândulas écrinas funcionam como uma resistência variável, que diminui consoante o aumento de transpiração ainda que a transpiração não seja visível. Esta medida, apesar de ser um bom indicador de stress, raiva, medo ou conflito, tem o problema de poder ser influenciada por fatores externos como por exemplo a temperatura ambiente. De forma a resolver este problema é necessário utilizar medidas de referência e calibração [MA07, ?].

Pulso de Volume Sanguíneo BVP⁶ é uma medida que permite determinar a quantidade de sangue que está a correr pelas veias. A medida é efetuada através de uma fonte de luz e um sensor que mede a quantidade dessa luz que é refletida no sangue, que poderá ser maior ou menor consoante a quantidade que está a fluir. O BVP permite portanto medir a amplitude dos pulsos de um indivíduo, o que significa que é possível também obter o ritmo cardíaco (HR⁷). Alguns dos problemas do uso do BVP devem-se ao facto de variar com a temperatura do meio ambiente em que o sujeito se encontra e por ser muito sensível à colocação e ao movimento, o que faz com que a comparação entre indivíduos não seja possível. [HGSW04, ?].

ECG (Eletrocardiografia) é uma medida mais precisa para medir o ritmo cardíaco através da atividade elétrica do coração [MA07]. Esta medição é feita com base no intervalo RR, que corresponde ao intervalo entre dois batimentos cardíacos. Este método é muito intrusivo por requerer a ligação dos sensores diretamente à superfície do peito do indivíduo, o que faz com que seja mais utilizado em circunstâncias clínicas. Um exemplo de um sensor usado em ECG pode ser observado na Figura 2.7.

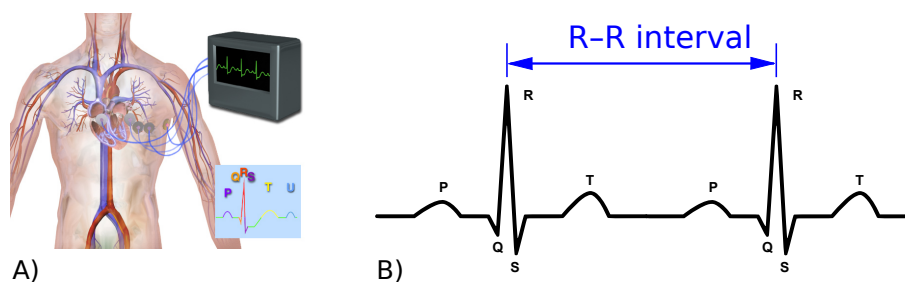


Figura 2.7: A) Uso de sensores numa ECG; B) Exemplo de intervalo RR⁸

Temperatura da pele mede simplesmente a temperatura na superfície da pele. Consoante a tensão dos músculos as veias sanguíneas podem contrair ou dilatar, o que altera a temperatura

⁴Do inglês: Galvanic Skin Response

⁵Do inglês: *skin conductance*

⁶Do inglês: *blood volume pulse*

⁷Do inglês: heart rate

⁸Fontes:

A) http://upload.wikimedia.org/wikipedia/commons/4/4d/Blausen_0339_Electrocardiogram.png

B) <http://upload.wikimedia.org/wikipedia/commons/c/c1/ECG-RRinterval.svg>

corporal. Esta medida tem alguns problemas tais como a dependência aos fatores externos (como no GSR) e a demora na indicação do estado emocional [HGSW04].

Respiração mede o quão depressa e profundamente um indivíduo está a respirar. A medida é calculada através da aplicação de uma faixa de borracha à volta do peito de um indivíduo. Uma respiração profunda e rápida indica emoções como medo, fúria e por vezes alegria. Já uma respiração pouco profunda e rápida é indicação de antecipação tensa que inclui pânico, medo ou concentração. Uma respiração lenta e profunda por outro lado indica um estado de relaxamento, enquanto que se for pouco profunda e lenta indica um estado de depressão ou felicidade calma.

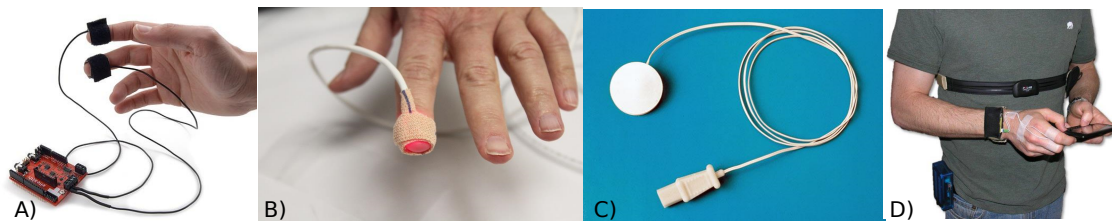


Figura 2.8: A) Sensor para SC; B) Sensor para BVP; C) Sensor para medir a temperatura da pele; D) Sensor para medir a respiração ⁹

2.3 Reconhecimento Emocional

Nas últimas duas décadas tem havido um aumento considerável na quantidade de artigos, com o intuito de criar e melhorar métodos que reconheçam emoções. Os métodos diferenciam-se em vários aspetos, tais como no número de emoções que são detetadas, sensores utilizados, algoritmos de classificação, espaços e modelos emocionais e ainda por serem efetuados em tempo real ou não. Por forma a melhorar a precisão do reconhecimento de emoções em trabalhos mais recentes, tem-se usado um maior número de sensores, o que permite aproveitar a extração das características dos sinais que melhor são recolhidos por cada um dos mesmos.

A título de exemplo da classificação de emoções específicas, no trabalho de Louis Schmidt et al. e Murugappan et al. [ST01, MRNY07] foi utilizada a EEG (Eletroencefalografia). No primeiro caso foram identificadas emoções como alegria, felicidade, medo e tristeza, através do uso de sensores ECG. Para classificar as emoções foi usado uma transformada discreta de *Fourier*. No segundo caso, foram identificadas as emoções felicidade, medo e repugnância através dos algoritmos FCM (*Fuzzy-C-Means*) e FKM (*Fuzzy-K-Means*), usando como característica de classificação a energia da emoção.

⁹Fontes:

- A) <http://www.cooking-hacks.com/blog/how-to-acquire-physiological-signals-with-arduino-and-measure-stress/>
- B) http://upload.wikimedia.org/wikipedia/commons/b/bd/Pulse_oximeter_sensor.jpg
- C) http://extww02a.cardinal.com/us/en/distributedproducts/ASP/STS-400.asp?cat=med_surg
- D) http://psi.cse.tamu.edu/wp-content/uploads/2012/07/wearable_sensors_slide.jpg

Mais tarde na investigação de Fatma Nasoz et al. [NLA03] foram identificadas emoções como tristeza, raiva, surpresa, medo, frustração e divertimento, através do uso de vários sensores: GSR, HR e de temperatura. Neste caso foram usados os algoritmos KNN (*K-Nearest-Neighbor*) e DFA (*Discriminant Function Analysis*), que permitiram classificar com precisão de respetivamente: raiva com 78,58% e 78,58%, tristeza com 75% e 90%, surpresa com 66,67% e 53,33%, frustração com 58,33% e 50% e divertimento com 43,75% e 56,25%.

Em 2006, Guillaume Chanel et al. [CKGP06] usaram o EEG, GSR, pletismografo para a pressão sanguínea, um cinto de respiração para avaliar os movimentos torácicos e abdominais e ainda um sensor de temperatura, com o objetivo de conseguir identificar corretamente a valência. O ritmo cardíaco foi calculado através da pressão sanguínea. Por forma a calcular a atividade da emoção foram apenas usados os sensores EEG que através do algoritmo *Naive Bayes* gerou resultados com uma eficácia média de 54% e uma eficácia média de cerca de 55% para o algoritmo FDA (Fisher Discriminant Analysis). Para os restantes sensores através do algoritmo FDA os autores conseguiram resultados com eficácia de cerca de 53% e finalmente para a fusão de características de todos os tipos de sensores, conseguiram-se resultados médios de 54%.

No trabalho de Regan Mandryk e Stella Atkins [MA07], foram detetadas emoções como diversão, desafio, aborrecimento, frustração e entusiasmo através do uso de algoritmos de lógica *Fuzzy*, num espaço AV.

Outra forma de classificar as emoções é através da distinção das emoções positivas, negativas ou neutras [Haz06, LCCS07]. Em 2007, Richard Hazlett [Haz06] usou a EMG facial. Neste processo foi utilizado um jogo ('Juiced') de onde foram gravados vídeos durante o período de jogo. Estes vídeos foram depois classificados como contendo eventos positivos ou negativos através dos sensores e depois através da opinião de três jogadores mais experientes. O acordo na classificação entre estes dois métodos foi de 100%. Depois a investigação de Enrique Leon et al. [LCCS07], foi das primeiras (dentro dos trabalhos estudados) a fazer uma distinção de emoções em tempo-real e independente do utilizador. Para conseguir que o sistema fosse independente do utilizador no entanto, o autor indica que é necessário ter conhecimento prévio da resposta fisiológica máxima e mínima para cada utilizador. Assim, neste trabalho, foram usados sensores que permitiram medir GSR, BVP e outras medidas extraídas a partir destas como por exemplo o HR. Através da utilização de algoritmos de ANN (*Associative Neural Network*), foi feita a distinção de emoções negativas, positivas e neutras com precisão de 71,4%.

O trabalho Cristina Conati e Heather Maclaren [CM09] de entre os estudados foi o único que usou o modelo OCC, por forma a caracterizar as emoções. Neste trabalho foi usada a EMG e uma *framework* que usa DDN (*Dynamic Decision Network*). Os autores deste trabalho usaram dois *datasets*, o primeiro *clear-valence* e o segundo *ambiguous-valence*, por forma a detetar as emoções: alegria, angústia, admiração e vergonha. Para o primeiro *dataset*, conseguiram valores de precisão de 79,10% para alegria, 56,70% para angústia, 83,18% para admiração e 63,02% para vergonha. Quanto ao segundo *dataset*, obtiveram 75,15% para alegria, 38,72% para angústia, 71,70% para admiração e 25,11% para vergonha.

No trabalho de Lindsay Brown et al. [BGP11] foram usados os sensores EEG, para calcular

três níveis de valência: positivo, negativo e neutro. Com o uso de imagens apenas, os autores conseguiram resultados com uma eficácia de 41% para três classes que é cerca de metade dos 82% obtidos quando foram usados apenas vídeos do IAPS.

Um outro trabalho que em vez do uso de sensores usou simplesmente a detecção de padrões na introdução de texto pelo teclado, foi o de Clayton Epp et al. [ELM11]. Neste trabalho foram detetadas emoções como confiança, hesitação, nervosismo, relaxação, tristeza e cansaço, com precisão a variar entre os 77,4 e os 87,8%. Para atingir estes resultados foram utilizadas árvores de decisão.

Mais recentemente nos trabalhos de Pedro Nogueira et al. [NRON13b, NRON13c] os autores têm-se focado em converter os sinais recebidos dos sensores de SC, EMG e BVP num sinal contínuo representado pelo espaço AV. No segundo [NRON13c], para obter este sinal foi usada uma arquitetura de duas camadas, composto por uma primeira camada em que são aplicados modelos de regressão e uma segunda, que usa árvores de decisão para fundir os sinais de saída da primeira camada. Com este método foi possível alcançar uma classificação de atividade e valência com respetivamente 89% e 84% de precisão.

Num outro trabalho de Pedro Nogueira et al. [NRON13b], tal como no trabalho referido anteriormente é uma vez mais utilizada arquitetura em duas camadas, no entanto neste caso em vez do uso de apenas árvore de decisão são utilizadas várias técnicas de *machine learning* (DT¹⁰, NN¹¹, RF¹², SVM¹³) que fundem os sinais de saída da primeira camada. Com este método, conseguiram-se resultados mais precisos, respetivamente de 97% para a atividade e 91% para a valência.

2.4 Geração Procedimental

Neste capítulo são apresentadas numa primeira fase diversas características inerentes aos métodos e técnicas de geração procedimental, seguidas de algumas técnicas e *frameworks* das mesmas.

2.4.1 Características

Os métodos de geração procedimental são dependentes do contexto de cada jogo [Hen12]. Como tal, não existe um método que permita gerar níveis para qualquer tipo de jogo, no entanto existem várias características inerentes a estes métodos que os *designers* devem ter em mente ao utilizá-los. Estes métodos podem ser assistidos ou não assistidos, *online* ou *offline*, construtivos, de geração e teste ou baseados em procura e podem ainda ser classificados como geradores de conteúdo necessário ou opcional.

Técnicas assistidas *versus* Técnicas não assistidas

¹⁰Decision Tree

¹¹Neural Networks

¹²Random forests

¹³Support Vector Machine

Aqui a principal distinção deve-se ao facto de haver ou não interferência humana no processo de geração. No caso das técnicas assistidas existe uma maior controlo, o que permite aos *designers* gerarem conteúdo mais apropriado a um determinado contexto. No entanto, estas técnicas têm a desvantagem de requererem mais tempo para parametrizar ou até ajustar o conteúdo, por forma a que este seja satisfatório. Por outro lado as técnicas assistidas, quanto muito requerem alguma parametrização o que permite a geração de cenas 3D com tamanho ilimitado e em tempo real. Estas técnicas podem ainda ser utilizadas para criar uma cena 3D base que pode depois ser trabalhada pela equipa de *design*.

No trabalho de Tim Tutenel et al. [TSL⁺11] foi usada uma aproximação mista, que tanto permite ao utilizador especificar o *design* como permite que a geração de conteúdo seja feita de uma forma completamente automática.

Técnicas *online* versus Técnicas *offline*

A principal distinção entre estas duas técnicas é que nas técnicas *online* o conteúdo é gerado em tempo real, enquanto que na outra o conteúdo é gerado na fase de desenvolvimento de jogo. No caso das técnicas *online*, uma vez que o conteúdo é gerado durante a sessão de jogo é necessário que sejam cumpridos alguns requisitos. Em primeiro lugar os métodos de geração têm que ser rápidos, uma vez que fazer o jogador esperar que o conteúdo seja gerado só tem desvantagens. Em segundo lugar é também necessário que os métodos de geração tenham um tempo de execução previsível. Finalmente, mas não menos importante, dependendo do contexto a qualidade do conteúdo gerado deverá também ser previsível. Um exemplo das técnicas *online* é quando um jogador entra dentro de um edifício e o interior do edifício é gerado momentaneamente [TYSB11].

Métodos construtivos versus Métodos de geração e teste

A principal diferença entre estes dois métodos é que nos métodos construtivos, existe um conjunto de regras que são usadas para gerar o conteúdo uma única vez. Por outro lado, nos métodos de geração e teste o conteúdo é primeiro gerado, depois são testadas as regras e no caso de haver conteúdo que não respeita as regras impostas, uma secção ou a totalidade da cena gerada no primeiro passo volta a ser gerada. No caso dos métodos construtivos, durante o processo de geração é feita a verificação de todas as regras, por forma a que o conteúdo gerado cumpra todos os requisitos.

Dentro dos métodos de geração e teste existem ainda os métodos baseados em procura que são um caso específico dos primeiros. Para um método ser considerado como sendo baseado em procura deve usar uma função de teste, que em vez de simplesmente aceitar ou rejeitar o conteúdo gerado, deve classificá-lo através de um número ou vetor de números reais. Para além desta restrição, ao gerar novo conteúdo o valor associado ao conteúdo gerado no passo anterior deve ser tomado em conta na geração de novo conteúdo [TYSB11].

Conteúdo opcional versus Conteúdo necessário

O conteúdo gerado não traz dependências ao jogador, isto é, o jogador não necessita deste conteúdo para poder avançar no jogo, sendo este conteúdo meramente decorativo. Por outro lado no conteúdo necessário, o jogador depende do conteúdo gerado para avançar no jogo, o que impõe

a restrição de este ter que estar sempre correto, pois não é aceitável um jogador não conseguir progredir devido a problemas presentes na geração.

Um exemplo de conteúdo opcional pode ser a geração de casas numa cidade, em que o jogador pode explorá-las, mas que não têm nenhum intuito para além de decoração. Por outro lado um exemplo de conteúdo necessário seria num jogo do tipo *dungeon*¹⁴, onde é necessário eliminar inimigos por forma a progredir no jogo. Nesse caso o inimigo teria que ser gerado numa zona onde o jogador tivesse acesso [TYSB11].

2.4.2 Técnicas

Serão agora apresentados algumas das técnicas de geração procedimental, que visam demonstrar algumas das técnicas que são aplicadas nos videojogos para criação de conteúdo, como por exemplo: terrenos, árvores, nuvens, entre outros.

Deslocamento do ponto médio

Em 2009, Xingquan Cai et al. [MCSL09] apresentam a técnica de deslocamento do ponto médio em duas dimensões, que é baseada em geometria fractal¹⁵ para criar modelos de elevação. Esta técnica permite criar um mapa de elevação que pode ser usado para criar terrenos acentuados.

Este método começa com uma área retangular (ABCD) representado na Figura 2.9) com atribuição de valores iniciais para a altura nos quatro cantos. Seguidamente, é calculada a altura do ponto E através da equação 2.1. A altura dos pontos F, G, H e I é calculada com base na equação 2.2. Os restantes pontos são calculados através da equação 2.3. O processo é repetido até que se atinja um nível de detalhe suficiente para o que é pretendido. A cada iteração a variável dH deve ser multiplicada por 2^{-r} , em que r é o valor da rugosidade do terreno.

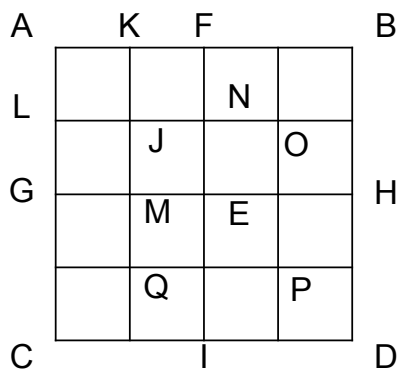


Figura 2.9: Deslocamento do ponto médio em duas dimensões

$$H_E = (H_A + H_B + H_C + H_D)/4 + R(dH, 1). \quad (2.1)$$

¹⁴Jogos geralmente compostos por níveis tipo labirinto

¹⁵figura de geometria não Euclidiana.

$$H_F = (H_A + H_B + 2 * H_E) / 4 + R(dH, 1). \quad (2.2)$$

$$H_J = (H_K + H_L + H_L + H_N) / 4 + R(dH, 2). \quad (2.3)$$

$$R(dH, 1) = rand(-1, 1) \cdot dH \cdot 2^{-1}. \quad (2.4)$$

A função R nas equações 2.1, 2.2 e 2.3 representa a função de geração de números aleatórios da equação 2.4. O parâmetro dH ($dHeight$) permite alterar a inclinação do terreno. Para valores de dH muito grandes poderão ser criadas regiões mais pontiagudas que para valores de dH pequenos.

Perlin Noise

O algoritmo de *Perlin Noise* tem como base uma função de ruído e uma função de interpolação. A função de ruído deve gerar números aleatórios e ser determinística.

Através da função de ruído são criados pontos aleatórios que são depois interligados através da função de interpolação. A função de interpolação pode ser linear, cúbica ou baseada na função cosseno [DZZ10].

Nos trabalhos de Wensheng Dong et al. e Mahmud Hasan et al. [DZZ10, HKA05] esta técnica foi utilizada para gerar nuvens no mundo virtual enquanto que no artigo de Wang Shunli et al. [WKW13] esta técnica foi utilizada em conjunto com outra para gerar ondas.

L-Systems

Um *L-System* ou sistema de *Lindenmayer* [PL90] consiste no uso de uma gramática e de um interpretador. A gramática contém um axioma que é usualmente um conjunto de caracteres pequeno, expandido pela gramática para um conjunto de caracteres longo e complexo. O interpretador converte depois o conjunto de caracteres num objeto [AGB05]. Num primeiro passo a gramática do sistema de *Lindenmayer* gera uma palavra através da aplicação recursiva de regras a um axioma. Num segundo passo, através da associação de uma determinada ordem geométrica a cada letra da palavra resultante, é efetuado um desenho.

No trabalho de Daniel Ashlock et al. [AGB05] este sistema é utilizado para gerar terrenos. Para este efeito é usada a técnica de *midpoint L-system* que é semelhante à técnica de deslocamento do ponto médio dos fractais, apresentada anteriormente. Baele [BW05] por outro lado faz uso deste sistema para gerar árvores.

Na investigação de Ricardo Lopes et al. [LB11b] após um levantamento dos vários métodos de adaptação de jogos, os autores concluem que os métodos de geração procedimental em combinação com a modelação semântica fornecem uma excelente ferramenta para a criação de conteúdo personalizado *offline* enquanto que a nível *online* fornecem uma excelente ferramenta para ajustar mundos, cenários e missões.

Gramáticas Geradoras

De acordo com o trabalho de Roland van der Linden et al. [vdLLB14] as gramáticas geradoras criam frases através de uma seleção finita que é feita de uma lista de regras de produção, que incluem palavras como símbolos terminais¹⁶. Existem algumas gramáticas que foram criadas a partir das gramáticas geradoras, como é o exemplo das gramáticas baseadas em nós e das baseadas em formas. Um exemplo do uso destas duas últimas gramáticas está presente no trabalho de Pedro Silva et al. [SMBC13].

Um exemplo de gramáticas geradores é apresentado no subcapítulo seguinte, onde é descrito o trabalho de Roland van der Linden et al. [vLLB13].

Algoritmos Genéticos

Os algoritmos genéticos são algoritmos evolucionários baseados em procura, que tentam encontrar uma solução ótima para um problema [Gol89]. O algoritmo genético tem dois requisitos: uma representação genética e uma função de *fitness*. A representação genética faz uso de strings denominados genes ou cromossomas para identificar possíveis soluções. A função de *fitness* por sua vez, é usada para avaliar a qualidade das soluções geradas. O método é iterativo e consiste no cálculo de valores de *fitness* para um conjunto de genes (população) e na geração de uma nova população. A geração da nova população tem sempre como base a iteração anterior e pode ocorrer por exemplo através do uso das técnicas de *crossover* ou mutação. No caso de *crossover* são usados dois genes "pais" para criar um filho, enquanto que no caso da mutação os genes são simplesmente alterados através da substituição de um carácter na sua representação genética. As mutações asseguram que em tempo indeterminado será sempre encontrada uma solução ótima [vdLLB14].

Uma aplicação dos algoritmos genéricos à geração procedimental, pode ser encontrada no trabalho de Manuel Kerssemakers et al. [KTTY12] onde o algoritmo é usado para gerar um gerador de conteúdo.

Autómato Celular

Um autómato celular consiste numa estrutura que se organiza automaticamente e iterativamente ao longo de um período de tempo. A estrutura consiste numa grelha de células com um número finito de linhas e colunas. Cada célula da grelha tem um estado inicial (quando tempo = 0). Para calcular o estado da célula ao longo do tempo, é executado um conjunto de regras que têm em conta o estado atual assim como o estado das células vizinhas [vdLLB14].

Um dos exemplos mais práticos do uso deste tipo de algoritmo é o "Game of Life" que foi concebido pelo matemático britânico John Horton Conway em 1970. Segue-se abaixo um exemplo de uma sequência do algoritmo na figura 2.10 [Kar11]. Este exemplo utiliza as seguintes duas regras em cada iteração:

¹⁶Símbolos que não podem ser modificados pela gramática

1. Se a célula está viva¹⁷, matém-se viva se e só se dois ou três dos seus vizinhos estiverem vivos. Caso contrário morre por solidão ou excesso de população.
2. Se uma célula está morta¹⁸ então passa a estar viva se e só se exatamente três dos seus vizinhos estiverem vivos.

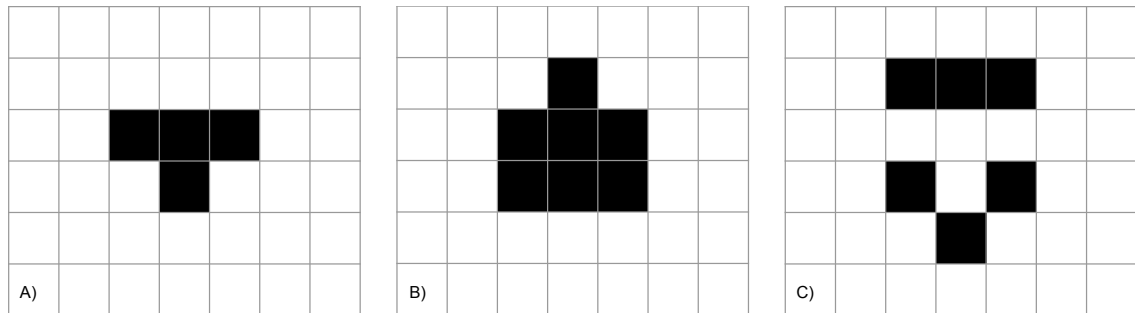


Figura 2.10: Representação de três iterações do algoritmo Game of Life: A) tempo = 1; B) tempo = 2; C) tempo = 3;

2.4.3 Frameworks

Nesta secção serão apresentadas algumas *frameworks*, sobretudo relativas a adaptação de jogos, uma vez que estão mais dentro do tema desta dissertação.

No trabalho de Ricardo Lopes e Rafael Bidarra [LB11a] os autores por forma a tornar a experiência de jogo personalizada e os jogos mais divertidos e menos previsíveis, criaram uma *framework* que tem como objetivo gerar ou alterar o conteúdo do jogo baseado no desempenho do jogador. Os princípios arquitetónicos que são a base desta *framework* são descritos na figura 2.11.

A adaptação dos jogos depende de uma metodologia de dois passos: a modelação do jogador e a geração de conteúdo. O que significa que a ligação entre as duas não só é natural como obrigatória para que a adaptação funcione.

No que toca à modelação do comportamento do jogador, existem já várias técnicas propostas que pretendem modelar alguns aspetos do seu comportamento. Uma das técnicas é a aprendizagem máquina que foi já usada para modular as habilidades do jogador em jogos de plataformas e tiros e para modular as preferências em jogos de estratégia. Para além das técnicas de aprendizagem máquina já foram propostas outras técnicas que são construídas manualmente e que combinam métricas do jogador e características do modelo associadas.

Relativamente à modelação da experiência foram identificadas três abordagens: subjetiva, objetiva e baseada na jogabilidade. As três abordagens distinguem-se essencialmente nos dados do jogador em que se baseiam para avaliar e caracterizar a jogabilidade. A primeira (subjetiva), através de dados fornecidos pelos jogadores (por exemplo: auto-avaliações), a segunda através de

¹⁷célula representada a preto

¹⁸célula representada a branco

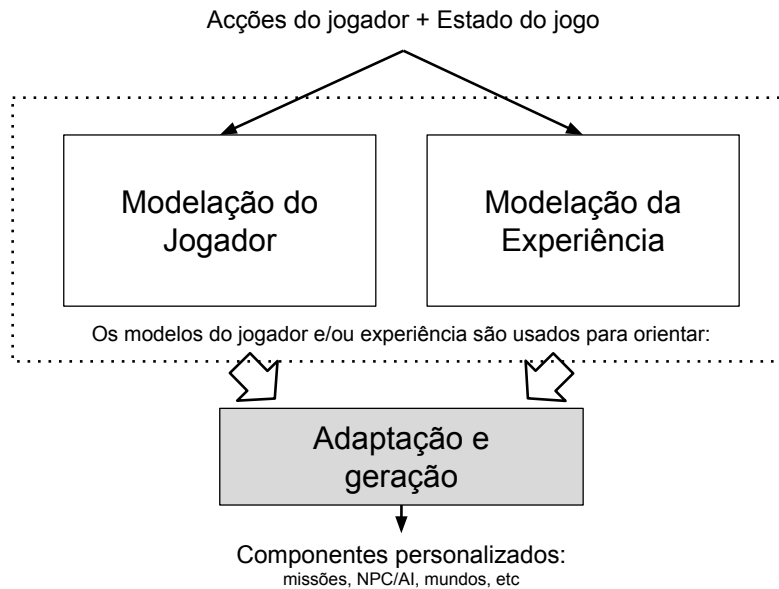


Figura 2.11: Vista geral da arquitetura dos princípios da adaptabilidade de jogos

recolha de vários tipos de dados de entrada (por exemplo: seguimento de movimentos e do movimento dos olhos) e por último através de dados obtidos através da interação do jogador com o jogo, como por exemplo métricas do jogo.

Na continuação do trabalho realizado por Ricardo Lopes et al. [LB11a], Ricardo Lopes et al. [LTB12] criaram a *framework* de geração semântica que é responsável por integrar a modelação do comportamento e da experiência do jogador com a geração procedimental de conteúdo, possível através do uso da semântica do mundo do jogo. Nesta *framework* a semântica de jogo é definida pelos *designers* através de uma biblioteca de edição de semântica que serve esse propósito. Por forma a facilitar o trabalho dos *designers* e a tornar esta tarefa menos tediosa, foram criados alguns mecanismos para ajudar na definição das mesmas.

Com esta ferramenta, os autores concluíram que não só é possível criar mundos adaptados ao comportamento e experiência do jogador, como ainda é possível obter a variabilidade esperada da geração procedimental.

Mais recentemente Roland van der Linden et al. [vLLB13] apresentaram uma ferramenta que permite aos *designers* especificarem as restrições de jogo através da especificação das ações do jogador. Estas ações descrevem a jogabilidade através das indicações daquilo que um jogador pode fazer num nível. No entanto, como não existe nenhum conjunto de ações universal, é importante que para cada tipo de jogo sejam os *designers* a fazê-lo. Por forma a facilitar a indicação de atributos semânticos e relações em forma de restrições a serem utilizadas para resolver possíveis conflitos durante o processo de geração, foi usada a biblioteca de semântica Entika.

As ações são especificadas através de pares de verbo-objeto, como por exemplo "matar o dragão". Os alvos estão geralmente relacionados com conteúdo do jogo, como é o exemplo do

dragão na ação "matar o dragão". O conteúdo do jogo neste caso refere-se às personagens não jogáveis e às relações entre elas. As ações podem ainda ser agrupadas de forma a serem mais interessantes e a terem uma maior usabilidade.

O caso de estudo deste trabalho foi o jogo *Dwarf Quest*. Um exemplo do uso desta implementação com o caso de estudo pode ser observado na figura 2.12.

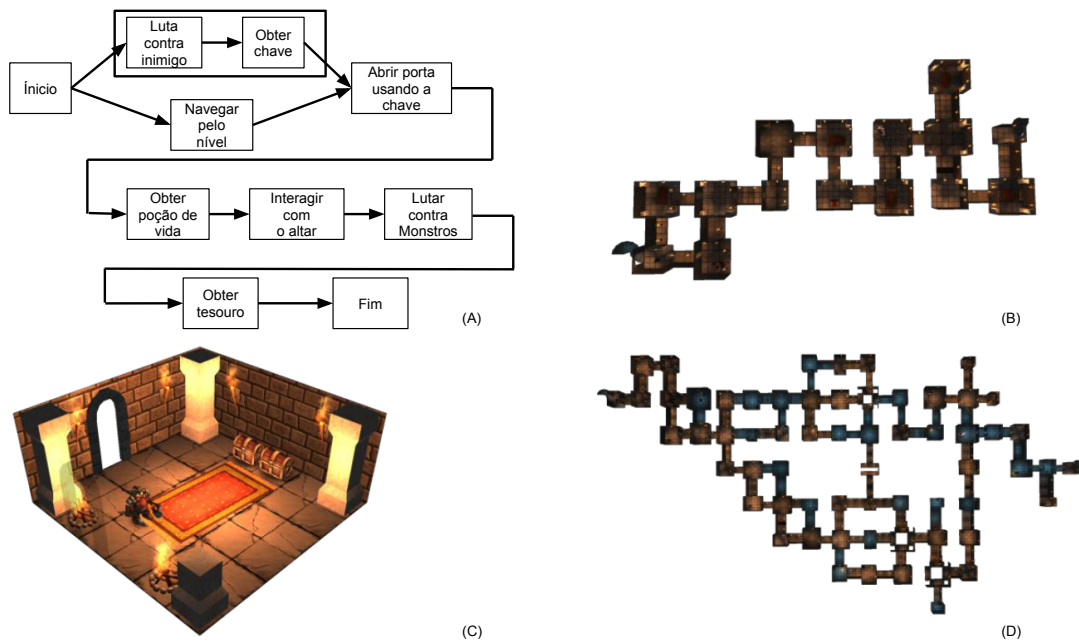


Figura 2.12: a) Grafo de ações do jogador para um jogo do tipo masmorra (*dungeon*); b) plano de uma masmorra gerada proceduralmente; c) Sala de recolha do tesouro; d) outro exemplo de uma masmorra gerada proceduralmente; Fonte: [vLLB13]

2.5 Geração Procedimental Afetiva

A aplicação da componente afetiva à geração de conteúdo procedimental é uma área relativamente recente. Com base nesta afirmação, seguem-se os trabalhos estudados nesta área, que têm como principal objetivo perceber como a componente afetiva de um jogador pode ajudar a alterar os algoritmos de geração procedimental.

No trabalho de Kiel Mark Gilleade et al. [GDA05] foi proposto um conjunto de três heurísticas denominado de ACE que pretendem melhorar o processo de desenvolvimento de jogos, através do uso de emoções. Os três modelos são *assist me*, *challenge me*, e *emote me*.

Assist Me

Este modelo, tem como objetivo principal evitar a frustração dos jogadores ou a incapacidade destes de progredirem num determinado nível, por não conseguirem atingir determinados objetivos. O autor propõe que através das medidas fisiológicas, seja medida a frustração

do jogador, que em combinação com o conhecimento do contexto do jogo, deve identificar as situações onde podem ocorrer estes problemas e assim ajustar alguns aspetos do jogo, por forma a resolver os mesmos. Alguns exemplos de resolução dos problemas encontrados, passam por fornecer pistas ao jogador assim que o nível de frustração seja alto, para que este consiga superar os problemas encontrados.

Challenge Me

Este modelo, tem como principal objetivo resolver o problema de o nível de dificuldade de um jogo não se adequar a um jogador. A proposta de resolução para este problema passa por usar o nível da atividade do jogador para alterar dinamicamente o nível de dificuldade que o jogo proporciona, por forma a se adaptar melhor a cada jogador.

Emote Me

Neste último modelo, o principal problema passa pelo facto de um jogador poder não ter a experiência que o *designer* tinha em mente, por não conseguir sentir as emoções que haviam sido propostas. Como solução a este problema, deve ser feita uma deteção do estado emocional do jogador enquanto este se encontra a jogar, por forma a que o conteúdo seja alterado e as emoções pretendidas sejam as realmente sentidas pelo jogador.

Mais tarde no trabalho de Eva Hudlicka [Hud08] a autora implementou um sistema em que as personagens do jogo mudavam o seu comportamento afetivo dinamicamente. A implementação deste sistema tem por base o modelo de afetividade do jogador e a resposta do mesmo com o desenrolar do jogo.

Os modelos de afetividade são estruturas que guardam informações sobre o estado afetivo do jogador como por exemplo, que estímulos fazem com que o utilizador sinta determinadas emoções, que comportamentos estão associados a diferentes emoções, entre outros.

Em 2011, Georgios N. Yannakakis e Julian Togelius [YT11] criaram uma *framework* que permitiu uma aproximação à geração procedimental baseada em emoções. Para isso foram identificadas três classes de aproximações possíveis para modelar a experiência do jogador: dados exprimidos pelos utilizadores (subjativa), dados obtidos através de tipos/modalidades alternativas de resposta do jogador (objetiva) e dados obtidos através da interação entre o jogador e o jogo (baseado na jogabilidade).

Na modelação subjativa os dados são recolhidos através de questionários obtidos diretamente através dos jogadores ou através da análise da fala ou escrita do jogador ao jogar.

Na modelação objetiva os dados são obtidos através de eventos comportamentais que ocorrem durante sessões de jogo, como por exemplo alterações nas expressões faciais, postura ou fala. Para alcançar este tipo de modelação podem ser usadas medidas e ferramentas como a EMG, ECG, entre outras.

Na modelação baseada na jogabilidade os dados são obtidos através das ações dos jogadores, através da análise de padrões de interação.

De entre os três modelos o último é o mais eficiente a nível computacional e o menos intrusivo, no entanto em geral resulta em modelos com pouca informação que dependem de fortes suposições.

Tendo obtido o modelo de experiência do jogador, o passo seguinte passa por gerar o conteúdo e avaliá-lo. Para avaliar o conteúdo existem três aproximações possíveis, funções de avaliação direta, funções de avaliação baseadas em simulação e funções de avaliação interativas.

As funções de avaliação direta extraem algumas características do conteúdo gerado e mapeiam estas características diretamente em valores de qualidade. Estas funções podem ser baseadas em teoria ou em dados. Nas funções baseadas em teoria, o *designer* é guiado pela intuição, por teorias da emoção ou ainda pelas emoções do jogador. Por outro lado, nas funções baseadas em dados é feita uma recolha de dados em vários exemplos de conteúdo através de questionários, medidas fisiológicas, etc. Seguidamente através dos dados recolhidos é feito um mapeamento para emoções do jogador e finalmente para as funções de avaliação.

As funções de avaliação baseadas em simulação usam agentes de inteligência artificial para jogar pelo conteúdo gerado. Deste modo é possível por exemplo verificar se é possível atravessar um labirinto, etc. As funções baseadas em simulação podem ainda ser estáticas ou dinâmicas. No primeiro caso os agentes não devem mudar ao jogar. No segundo o agente vai alterando o seu comportamento durante o jogo, podendo por exemplo usar algoritmos de aprendizagem para avaliar o conteúdo em termos de velocidade de aprendizagem, isto é, o tempo que um utilizador deverá necessitar até aprender como funciona o jogo.

As funções de avaliação interativa são funções que avaliam o conteúdo com base na interação do jogador com o conteúdo. Esta avaliação pode ser feita de forma explícita, através de questionários ou outros métodos, ou de forma implícita através das ações do jogador e da forma como este interage com o mundo virtual.

Através do modelo da experiência e da função de avaliação, é depois possível otimizar o conteúdo gerado através da maximização dessa função.

Mais recentemente Pedro A. Nogueira et al. [NRON13a] propõem a *framework* E^2 que tem como principal função usar emoções para modelar a experiência afetiva do jogador.

Para isso, a *framework* foi dividida em várias componentes. A primeira componente é o *PI-ERS*¹⁹ que é responsável por classificar o estado emocional do utilizador no espaço AV. Outra componente é o *ARE²S*²⁰, que é responsável por determinar quais os mecanismos que devem ser ativados, para que seja transmitida uma determinada emoção ao jogador. A determinação dos mecanismos a serem ativados é baseada no histórico das reações do jogador. Para além do *ARE²S* é também utilizado o *ARC*²¹, que é uma base de dados que contém um conjunto de perfis de reação afetiva para cada utilizador.

¹⁹do inglês: Physiologically-Inductive Emotion Recognition Sub-system

²⁰do inglês: Affective Reaction Extraction and Extension Sub-system.

²¹do inglês: Affective Reaction Compendium

Para complementar o *ARE²S* é ainda usado o *CLEAR²²*, que é responsável pela monitorização e provocação de determinados estados/padrões emocionais. Por forma a ativar os eventos é usado o *GLaDOS²³* que é uma interface genérica usada para esse efeito e o *IMP²⁴* que consiste no conjunto de eventos que podem ser ativados para influenciar o utilizador.

No trabalho de Nogueira et al. [NARO14], é descrita uma forma de modelar as reações emocionais dos jogadores através de dados fisiológicos. Os dados fisiológicos são obtidos através da extração de características a partir das respostas emocionais dos utilizadores a eventos de jogo.

Desta forma os autores apontam à criação de modelos individuais para cada jogador por forma a prever as respostas emocionais deste a cada evento de jogo.

Através deste trabalho, os autores esperam que este método seja usado não só para jogos afetivos mas também para otimizar jogos que não o sejam antes que estes sejam produzidos, por forma a testar depois com jogadores (humanos) o conteúdo gerado.

2.6 Outros Trabalhos Relevantes

Um trabalho relevante é o trabalho Luís Teófilo et al. [TNS13] em que foi apresentada uma *framework* que permite simular o input do teclado através do input de vários tipos de controladores de jogos, como é o exemplo do WiiMote™, Kinect™ e do Rock Band™. O principal objetivo deste trabalho é o de criar uma interface que permita usar estes controladores em jogos que não estão preparados para o fazer. Um dos exemplos apresentados foi o uso destes controladores para detetar movimentos por forma a controlar a personagem do jogo The Elder Scrolls V: Skyrim.

A razão pela qual este trabalho é importante para o trabalho a ser desenvolvido é fato de o foco de ambos ser a modularidade, isto é, o facto de ser possível usar estas metodologias em jogos que não estão preparados ou a jogos a serem desenvolvidos futuramente. Outro aspeto que estes dois trabalhos têm em comum é o facto de ambos pretenderem melhorar a experiência de jogo. Por outro lado, enquanto que na metodologia a ser apresentada neste trabalho esta melhoria ocorre através do uso dos recursos já disponíveis no jogo (exceto os sensores), no outro são usadas novas ferramentas que pretendem fazer com que o jogo se torne mais divertido.

Outro trabalho relevante é o trabalho de Ambinder [Amb11] em que são discutidos vários métodos de obtenção dos dados afetivos. Na discussão apresentada são apresentadas vantagens e desvantagens de cada um dos métodos apresentados, desde os sinais fisiológicos a expressões faciais, movimento dos olhos, entre outros. Depois deste levantamento foram ainda mostrados os efeitos da aplicação dos dados afetivos ou do uso dos sensores como controladores em jogos como Left4Dead2, Alien Swarm e Portal 2. Para tal, foram feitas alterações nos motores de controlo da inteligência artificial do Left4Dead2, para o Alien Swarm foram usados os dados afetivos como

²²do inglês: Closed-Loop Emotional Adjustment and Regulation Sub-system

²³do inglês: Game Layer alteration Daemon Operating Script

²⁴do inglês: Implicit Mechanism Pool

dados de entrada para funções que faziam alterações no jogo e finalmente para o portal 2 foi usado o movimento dos olhos para controlar a personagem de jogo.

2.7 Resumo

Apesar de só há relativamente pouco tempo ter começado a haver um maior interesse em usar as emoções em sistemas de informação, as teorias de emoção têm já vindo a ser alvo de estudo há muito mais tempo. Nos trabalhos estudados foram usadas várias teorias da emoção que passaram pelo espaço AV, o espaço OCC, a teoria de Plutchik e o cubo de Lövheim.

Em relação aos sensores e aos métodos de reconhecimento de emoções, têm-se conseguido identificar corretamente as emoções com precisões relativamente altas o que é um bom indicador para o uso do reconhecimento de emoções aplicado à interação pessoa computador. A deteção de emoções em tempo real, é algo que também se tem mostrado ser possível fazer com precisão cada vez mais próxima das outras aproximações.

Quanto à geração procedimental, têm havido grandes progressos nos últimos anos e tem havido acima de tudo uma grande preocupação em tornar estes métodos mais simples e interativos.

Finalmente, em relação à geração procedimental afetiva é possível notar uma tendência de crescimento nos últimos anos.

Revisão Bibliográfica

Capítulo 3

Metodologia

Neste capítulo é descrita a metodologia desenvolvida com vista à solução do problema descrito na introdução desta dissertação. Numa primeira fase é apresentada a arquitetura seguida da descrição dos vários componentes e das suas interligações, as quais se centram sobre o motor de regras que permite a definição e teste flexível de condições de jogo. Finalmente são apresentados alguns casos de aplicação possíveis.

Esta metodologia foi desenvolvida com vista a resolver alguns dos problemas que existem na indústria dos jogos, pelo que o seu desenvolvimento foi de encontro a algumas propriedades que pretendem torná-la mais aceite pela indústria:

Fácil implementação: deve ser fácil implementar esta metodologia para incentivar as produtoras de jogos a aplicarem a mesma nos seus jogos, sem necessidade de grandes investimentos quer a nível de tempo quer a nível financeiro;

Genérica: deve ser possível aplicá-la na maior parte dos jogos, quer estes já tenham sido desenvolvidos ou estejam ainda na fase de desenvolvimento;

Reutilizável: após a implementação da metodologia para um caso específico, não deve ser necessário muito esforço para fazer alterações por forma a adaptá-la a um novo jogo, ainda que este seja de um tipo diferente do qual foi implementado.

3.1 Arquitetura

Existem três componentes (figura 3.1) que compõem a arquitetura desta metodologia são: o jogo, a *framework* e uma componente que permite a obtenção de dados afetivos.

O principal propósito da divisão da arquitetura por estas três componentes é o de a tornar reutilizável.

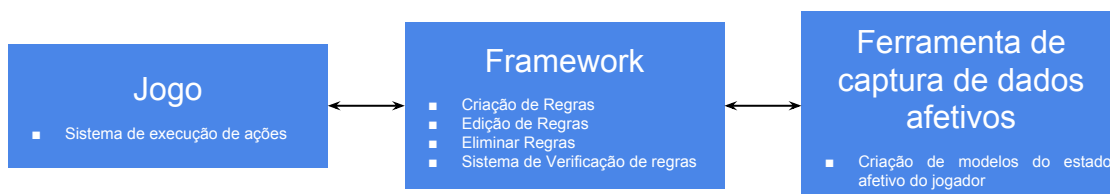


Figura 3.1: Arquitetura da solução proposta

O jogo é a componente básica sem a qual esta metodologia não faria sentido. É no jogo que são feitas as alterações que permitem aos *designers* testarem as suas criações. É ainda a componente onde haverá uma maior interação por parte do utilizador.

A *framework* é a componente que trata de todos os aspetos relacionados com as regras. É nesta componente que deve ser possível criar, editar e remover regras. Para além destas operações sobre as regras é na *framework* que devem ser efetuadas todas as verificações das condições das regras e posteriormente executadas as ações que são transmitidas ao jogo.

A aplicação de obtenção de dados afetivos é a aplicação onde são tratados os dados recebidos pelos sensores ou onde são criados e processados os modelos de estado afetivo do jogador.

3.2 Componentes

Comunicação

As três componentes descritas na arquitetura têm que conseguir comunicar entre si. Esta comunicação deve ser bidirecional e assíncrona. A comunicação deve ser bidirecional porque é necessário haver uma troca de informação constante entre as três componentes. Deve também seguir preferencialmente um modelo do tipo cliente-servidor em que a *framework* é o servidor, enquanto que o jogo e a componente de obtenção de dados afetivos são clientes.

Framework

A *framework* tem que ter acesso ao estado do jogo e ao estado afetivo do jogador. As alterações no estado do jogo ou no estado afetivo do jogador devem servir como mecanismo de ativação do sistema de verificação e execução das regras. Com base neste mecanismo de ativação e uma vez que a taxa de atualização de ambos os estados é na maior parte das vezes diferente, é necessário que o *designer* tenha possibilidade de escolher para cada regra qual dos estados deve ser usado como mecanismo de ativação.

É necessário o uso de uma *thread* por cada ligação às aplicações cliente (jogo e ferramenta de obtenção de dados afetivos) por forma a que a interferência com o sistema de verificação e execução de regras sofra a mínima interferência possível.

A *framework* tem ainda uma componente de *parsing* de mensagens que serve para processar as mensagens recebidas e envias pelo jogo e pela aplicação de obtenção de dados afetivos, servindo depois para manter em memória uma cópia do estado de jogo e afetivo atuais.

Uma outra componente que é das mais importantes na *framework* é a interface de interação com o utilizador (*designer*) que pode ou não ser gráfica. Esta interface deve permitir a criação, edição e eliminação de regras em tempo real, o que oferece aos *designers* uma ferramenta de prototipagem rápida e flexível, o que lhes permite obter em tempo real uma ideia de como pequenas ou grandes alterações em algumas regras influenciam o estado afetivo do jogador.

Por forma a que a *framework* seja o mais modular possível, o sistema de regras é construído com base num conjunto de especificações que são declaradas através de meta-dados, o que permite uma maior manutenção e modulação de novos tipos de objetos, variáveis, entidades, etc.

Jogo

No jogo deve ser implementada toda a lógica de comunicação com a *framework* e de verificação do estado de jogo. Por forma a garantir que a jogabilidade não é afetada a comunicação deve implementada numa *thread* à parte. Por forma a reduzir o número de mensagens enviadas ao servidor, o estado do jogo deve apenas ser transferido se existirem alterações no mesmo. Ainda assim, se possível devem apenas ser consideradas as variáveis que foram alteradas e não toda a estrutura. Para a comunicação deve ainda ser implementado um *parser* que trate das mensagens recebidas pela *framework* e execute as ações pretendidas no jogo.

Para verificar se a estrutura de jogo sofreu alguma modificação, pode também ser criada uma *thread* à parte (uma vez mais de modo a afetar o mínimo possível a jogabilidade) ou pode ser implementado um sistema de manipulação de eventos que gera uma notificação cada vez que uma das variáveis do estado de jogo é alterada. Para o caso de um jogo que já foi concretizado, será possivelmente mais fácil implementar uma *thread* à parte que compare o estado de jogo atual com um estado de jogo anterior, por forma a verificar se existem alterações.

Obtenção de dados afetivos

Embora a componente de obtenção de dados possa surgir em qualquer uma das outras duas componentes é recomendado que seja feita em separado por forma a facilitar a manutenção e futuras alterações. Por exemplo, inicialmente os dados afetivos podem ser determinados através do comportamento do utilizador (através dos cliques no rato, teclas premidas no teclado, etc.) sendo possível mais tarde o uso de sensores para obter os dados fisiológicos que permitem obter uma maior precisão.

No caso de esta componente ser implementada independentemente das restantes, é necessário implementar a lógica de comunicação com a *framework*. Por outro lado se a componente for desenvolvida juntamente com a *framework*, esta deve ser executada numa *thread* à parte de maneira a não bloquear a interface da mesma.

É a partir desta aplicação que devem ser armazenados e visualizados os dados afetivos por forma a permitir aos designers ajustarem rapidamente as regras, para conseguirem transmitir corretamente os estados emocionais pretendidos.

3.3 Regras

As regras são compostas por conjuntos de condições e de ações. As condições podem ser simples ou compostas e as ações podem ser simples ou complexas.

Condições simples: são condições que têm como objetivo verificar apenas se uma variável, avaliação de uma função, resposta a um evento ou outro tipo de evento respeita uma condição lógica, como por exemplo verificar se o nível de vida de uma personagem é maior que 50%;

Condições compostas: são condições que são formadas por condições simples e são unificadas através das funções lógicas: negação, “E”, “OU”, entre outras. Estas condições permitem criar regras mais complexas, que têm como objetivo alargar o número de possibilidades da especificação de regras.

Ações simples: são ações que permitem alterar diretamente uma variável, chamar uma função, etc. Alguns exemplos mais concretos podem passar pela alteração da iluminação, pela geração de uma nova porção de um nível ou pela geração de alguns tipos de *items*.

Ações complexas: são ações que são compostas por ações simples. No entanto, não passam apenas por repetir um determinado conjunto de ações, mas sim por executar ações que sigam algumas regras. Alguns exemplos de ações deste tipo são as que se seguem:

Ações temporizadas: Ações que são executadas ao longo de um determinado período de tempo. Neste caso a ação poderá ter um *delay* inicial e decorre durante um determinado número de iterações separadas por um intervalo de tempo bem definido;

Ações condicionadas: Ações que têm que ser executadas por uma determinada ordem e que dependem da execução prévia de outras ações. Podem ainda ser ações temporizadas que acabam consoante a verificação de uma condição em vez de ser após a conclusão do intervalo de tempo.

De modo a facilitar a construção de regras pode ser utilizado um editor de condições e ações globais, bastando depois fazer referências às mesmas. Desta forma é possível reutilizar certas condições ou ações sem ser necessário definir novamente cada uma delas.

Para resolver problemas inerentes das ações temporizadas existem duas soluções possíveis. A primeira solução passa por tratar deste tipo de ações no jogo, implementando um mecanismo de interrupção. Um exemplo específico de um jogo do tipo First-Person Shooter (FPS), seria na geração de inimigos durante dez iterações em que seria gerado um por iteração. Neste exemplo ao fim das dez iterações caso a personagem não eliminasse nenhum dos inimigos, passariam a estar dez inimigos a atacar o jogador. Faz portanto todo o sentido neste caso acabar ou interromper o sistema de geração, por forma a que o nível de dificuldade não se torne demasiado alto. Por outro lado, a segunda solução consiste em implementar este tipo de ações na *framework*. A primeira solução apresenta como principal vantagem o facto de ser necessário um menor número de

mensagens entre a *framework* e o jogo. Por outro lado, apresenta uma grande desvantagem ao ser necessário fazer as verificações no jogo, o que torna a arquitetura menos modular.

Uma vez que a *framework* é a principal componente que será utilizada por *designers*, que possivelmente não têm experiência a nível de programação, é importante que a especificação de regras seja intuitiva. Para isso poderá ser desenvolvido um sistema de declaração visual ou através de uma linguagem de alto nível. Apenas desta forma é garantido que o tempo dos *designers* é usado para a geração da parte visual e criativa e não na aprendizagem do funcionamento da ferramenta.

3.4 Aplicações

Após a definição da metodologia seguem-se agora alguns exemplos de como esta pode ser utilizada para determinados tipos de jogos, servindo para demonstrar algumas das potencialidades da metodologia.

No caso de um jogo de terror, é possível que os *designers* pretendam que o jogador sinta medo apenas em algumas partes do jogo. Neste caso o *designer* pode obter a partir do estado do jogo a posição do jogador e os seus dados afetivos, manipulando por exemplo através das regras a iluminação, o aparecimento de monstros, os sons, entre outras variáveis que tendem a afetar o sentimento de medo do jogador.

No caso de um jogo de corridas de carros em que o *designer* pretende que o jogador se sinta desafiado, mas não o tenha conseguido concretizar, pode através das regras gerar novos obstáculos que criem um maior desafio ao jogador. Podem ainda por exemplo alterar as condições climatéricas, imaginando que a ocorrência de chuva causa instabilidade e a perda de tração do carro.

3.5 Resumo

Neste capítulo foram apresentadas e descritas as componentes que formam um sistema capaz de utilizar o estado do jogo e os dados afetivos do jogador que juntamente com um conjunto de regras permitem aos *designers* modificarem o conteúdo gerado, com o objetivo de modelar esse conteúdo segundo as emoções que pretendem passar aos jogadores.

Foi explicado de que forma cada uma das componentes devem ser implementadas e ainda foram apresentados alguns problemas que podem surgir, assim como possíveis soluções para os mesmos.

Foram ainda apresentados alguns casos e formas em que esta metodologia pode ser aplicada.

Metodologia

Capítulo 4

Implementação

Neste capítulo será descrito em primeiro lugar o caso de estudo, que será seguido da apresentação da solução implementada assim como alguns dos seus aspetos mais técnicos.

4.1 Caso de Estudo

Por forma a demonstrar que a metodologia é genérica o suficiente para poder ser aplicada no maior número de jogos possíveis, foi escolhido um jogo *open-source* que foi alterado por forma a conseguir comunicar com a *framework*. Foi escolhido o jogo Eduke32¹ que é uma versão *open-source* do jogo Duke Nukem. A escolha do jogo deveu-se ao facto de ter uma fácil gestão de geração de inimigos, armas e outros *items*, devido à existência de uma linha de comandos interna.

O *Duke Nukem* é um jogo do tipo *First-person shooter* (FPS). A história passa-se no início do século 21, em Los Angeles (L.A.) após uma invasão extraterrestre. A polícia de L.A. sofreu uma mutação e as estradas da cidade tornaram-se numa zona infestada. Apenas uma pessoa pode salvar o planeta Terra e essa pessoa é Duke Nukem [Wil].

O jogo está dividido em quatro episódios, cada um composto por vários níveis, sendo ainda possível importação de níveis criados por fãs do jogo. Em cada nível o jogador deve recorrer às várias armas que tem à sua disposição para destruir os extraterrestres que encontra pelo caminho, até encontrar o botão (figura 4.1) de fim de nível. Para além das armas existentes ao longo do nível é possível obter vários *items* que ajudam a personagem a sobreviver. Em alguns dos episódios existem ainda para além dos inimigos comuns, inimigos com nível de dificuldade muito mais elevado.

¹Eduke32: <http://www.eduke32.com/>, data de último acesso: 18/06/2014



Figura 4.1: Botão de fim de nível (Duke Nukem 3D)

4.2 A Solução

Para implementar a metodologia foram criadas duas aplicações e foram feitas algumas modificações no jogo apresentado na secção anterior (4.1). As duas aplicações criadas foram a *framework* e a aplicação Sensor Reader, que serão melhor descritas ao longo deste sub-capítulo.

4.2.1 Arquitetura

A solução proposta é composta por três componentes que comunicam por sockets. Estas componentes podem ser identificadas nos retângulos a azul da figura 4.2, onde é apresentada a arquitetura da solução. A *framework* tem ainda três sub-componentes que estão identificadas nos retângulos a vermelho.

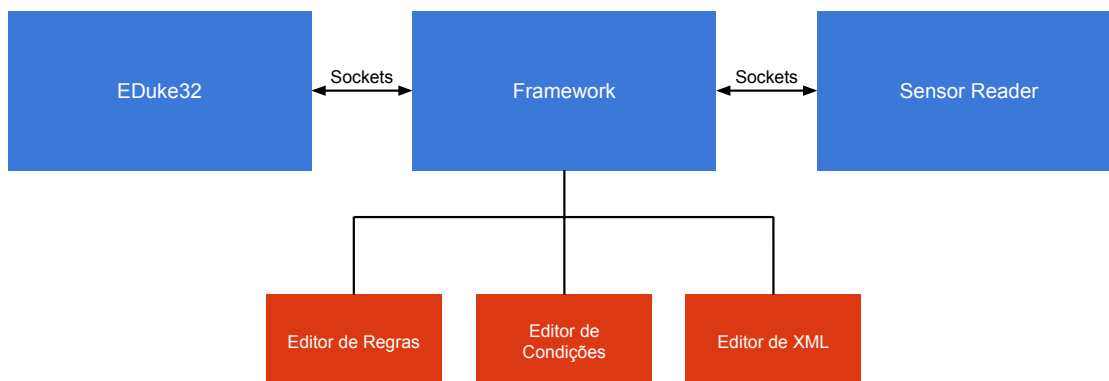


Figura 4.2: Arquitetura da solução

Implementação

As três componentes nos retângulos a azul correspondem à implementação das componentes já apresentadas no capítulo 3, que como indicado nesse mesmo capítulo foram implementadas separadamente por forma a facilitar possíveis alterações nos métodos de obtenção dos dados afetivos. Segue-se uma descrição mais detalhada daquilo que foi implementado em cada uma destas três componentes, por forma a adaptar a solução a este jogo.

Jogo: toda a lógica de comunicação assim como um *parser* de mensagens, para as mensagens recebidas pela *framework*;

Framework: editor XML para criar os documentos de especificação para as ações e condições. Editor de condições e ações globais, que usa os documentos XML (meta-dados) para definir os intervalos e nomes das variáveis. Editor de regras que usa as ações e as condições definidas pelo editor de condições e ações. Estas três componentes são as que estão representadas por retângulos vermelhos na figura 4.2 e serão explicadas com melhor detalhe na secção 4.2.3.

Sensor Reader: toda a lógica que permite a comunicação com a *framework*. Toda a lógica que permite a leitura dos dados dos sensores e o seu tratamento. Lógica necessária ao armazenamento dos dados recolhidos pelos sensores.

4.2.2 Protocolo de Comunicação

Como já foi descrito no capítulo da [metodologia](#), a comunicação foi implementada de forma assíncrona e é efetuada nos dois sentidos, entre a *framework* e o jogo e entre a *framework* e a aplicação Sensor Reader. Segue-se uma ilustração deste protocolo na figura 4.3, onde estão representados os tipos de mensagens transmitidos entre as diversas componentes.

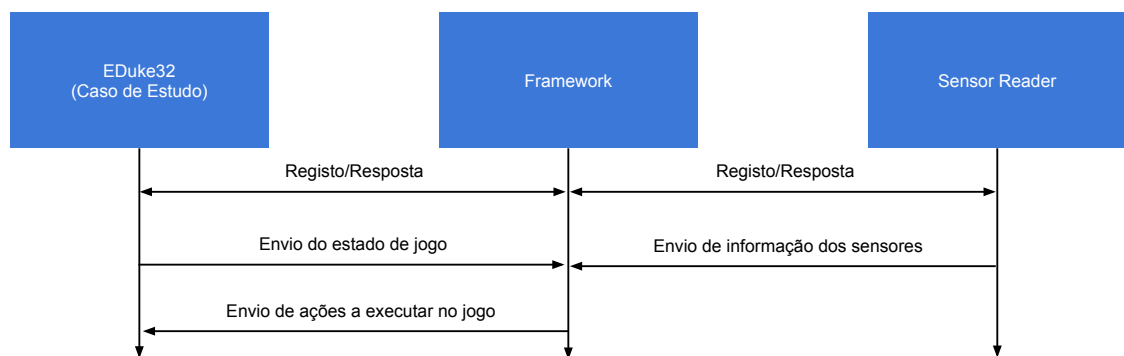


Figura 4.3: Protocolo de Comunicação

Numa primeira fase é necessário haver um registo de cada uma das aplicações cliente (EDuke32 e Sensor Reader) no servidor. Este registo garante que apenas uma instância do jogo e da aplicação de obtenção de dados afetivos está a comunicar com a *framework* de cada vez, evitando assim problemas de ligação futuros.

Implementação

Após o registo de cada uma das aplicações no servidor, este envia uma resposta a indicar que o registo ocorreu com sucesso. Finalizada esta etapa inicia-se um ciclo de comunicação que é apenas finalizado quando a *framework* ou o jogo terminarem. Este ciclo consiste na transmissão de dados entre as aplicações.

No caso da comunicação entre o jogo e a *framework*, a cada iteração o jogo envia à *framework* uma estrutura de dados que se traduz no estado do jogo. Por outro lado na comunicação entre os outros dois componentes os dados dos sensores são primeiro tratados pela aplicação Sensor Reader e são depois enviados para a *framework*.

A cada iteração em que os dados são recebidos, quer pela jogo quer pela aplicação Sensor Reader, ocorre uma verificação de todas as regras na *framework*. Para todas as regras cujas condições sejam avaliadas como verdadeiras, são enviadas as ações correspondentes para o jogo.

Por forma a diminuir o número de possíveis interrupções quer no jogo quer na *framework*, a comunicação é efetuada de forma assíncrona, tendo sido criadas duas *threads* quer no jogo quer na *framework*, que são usadas para o envio e receção de mensagens.

No anexo A.1 está representado um exemplo de mensagem transmitida entre o jogo e a *framework*, onde é enviado uma parte do estado do jogo.

4.2.3 Sistema de Regras

O sistema de regras é composto por três componentes, que estão indicados nos retângulos vermelhos da *framework* representados na figura 4.2.

Numa primeira fase é necessário usar um editor XML para criar as especificações XML. As especificações XML consistem na declaração de estruturas e ações possíveis no jogo. Tanto as ações como as estruturas são declaradas através das suas variáveis/parâmetros e dos intervalos/enumerações que essas variáveis/parâmetros podem tomar. Segue-se um exemplo de uma estrutura representada abaixo na listagem 4.1 que corresponde ao código gerado pelo editor de XML (figura 4.4). A tag XML “LowerBound” representa o valor menor do intervalo, enquanto que a tag “UpperBound” representa o maior. A tag “Enumeration” representa as opções possíveis.

Listagem 4.1 Declaração da ação Luz Ambiente.

```
1: <?XML version="1.0" encoding="UTF-8" standalone="no"?>
2: <Data>
3:   <Name>AMBIENT_LIGHT</Name>
4:   <Attribute>
5:     <Name>FUNCTION</Name>
6:     <Enumeration>R_AMBIENTLIGHT</Enumeration>
7:   </Attribute>
8:   <Attribute>
9:     <Name>VALUE</Name>
10:    <LowerBound>0.0</LowerBound>
11:    <UpperBound>10.0</UpperBound>
12:   </Attribute>
```

Implementação

The Data Editor interface is shown with the 'XML Editor' tab selected. The left sidebar contains a search bar and a list of actions: AMBIENT_LIGHT. (highlighted), GIVE_STUFF.xml, GODMODE.xml, REMOVE_ENEMIES.xml, SPAWN_ENEMIES.xml, SPAWN_HEALTH.xml, SPAWN_ITEMS.xml, and SPAWN_WEAPONS.xml. The main area displays the XML declaration for 'AMBIENT_LIGHT' with the following attributes:

Attributes/Parameters	Interval/Enumeration
FUNCTION	R_AMBIENTLIGHT
VALUE	[0.0,10.0]
TYPE	TYPE_CMDSETLIGHT

At the bottom of the main area, there are buttons for 'Add Row', 'Delete Row', and 'Help'.

Figura 4.4: Editor XML: Declaração da ação luz ambiente

```

13: <Attribute>
14:   <Name>TYPE</Name>
15:   <Enumeration>TYPE_CMDSETLIGHT</Enumeration>
16:   <Enumeration>TYPE_CMDSETVAR</Enumeration>
17: </Attribute>
18: </Data>

```

Após a criação das especificações *XML* é iniciada a fase de criação/edição de condições e ações. No caso das ações é necessário definir os valores para os parâmetros declarados na fase anterior, que terão obrigatoriamente que respeitar os intervalos/enumerações também definidos previamente. Já para as condições é apenas necessário definir para uma determinada estrutura, qual a condição que uma determinada variável deve respeitar (por exemplo: Vida do jogador maior ou menor ou igual a 50, figura 4.5).

The Condition/Action Editor interface is shown with the 'Condition/Action Editor' tab selected. The left sidebar contains a search bar and a list of conditions: AROUSAL < 0.8, Valence < 0.6, SECTOR 256, VALENCE <= 0.4, HEALTH <= 50 (highlighted), and BOSS_COUNT == 0. The main area displays the XML declaration for 'HEALTH <= 50' with the following attributes:

Attributes/Parameters	Value
Type	Game Objects
Object	Player
Parameter	HEALTH
Condition	<=
Value	50.0

At the bottom of the main area, there are buttons for 'Add condition' and 'Actions'.

Figura 4.5: Editor de condições e ações

Numa fase final, após a criação das condições e ações entra a fase de criação/edição de regras. Para criar uma nova regra, usando a interface gráfica, basta arrastar quer as condições quer as ações

Implementação

que estão representadas no lado direito para as áreas respetivas no lado esquerdo (ver 4.6). As condições podem ter uma estrutura mais complexa sendo possível agrupá-las. Se existirem mais que duas condições, o designer pode especificar quais as operações lógicas do tipo e/ou pretende realizar entre as mesmas. As ações também têm a possibilidade de serem várias, no entanto nesse caso serão todas executadas caso as condições especificadas se verifiquem. Segue-se um exemplo na figura 4.6 que representa a regra se vida do jogador menor ou igual a 50, então gera um item de vida.

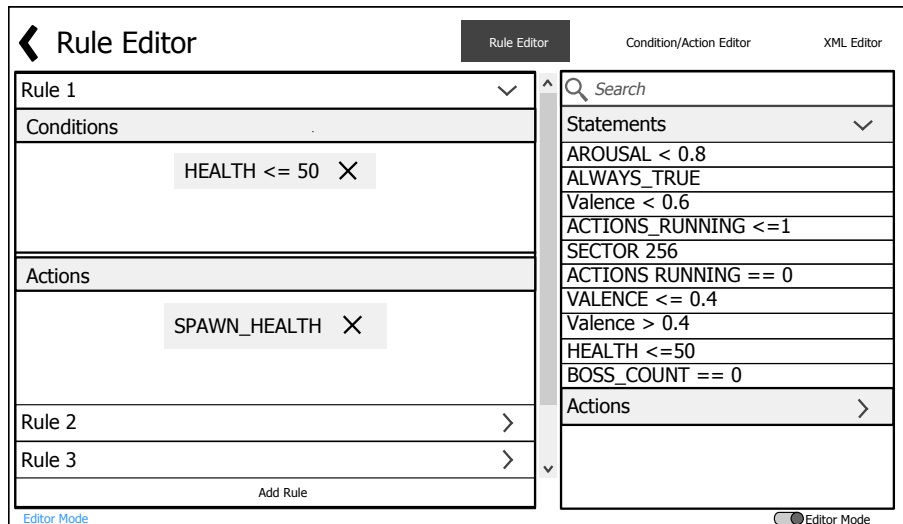


Figura 4.6: Editor de regras

4.3 Regras

Através do sistema de regras descrito na secção anterior foram criadas as especificações, condições, ações e regras que permitiram fazer alterações no caso de estudo.

No que toca à especificação, foram criados três grupos: ações, objetos de jogo e dados dos sensores.

Usando o editor de XML foram criadas oito especificações para ações, duas para objetos de jogo e 1 para dados dos sensores. Através destas especificações é depois possível criar as regras.

Segue-se agora uma lista do que as especificações criadas permitem executar no jogo:

AMBIENTLIGHT: Permite alterar a iluminação global do jogo;

GIVESTUFF: Gera pacotes de armas, munições, armadura, chaves, inventário que são depois fornecidas ao jogador;

GODMODE: Altera o modo de jogo para "GOD MODE"o que faz com que a personagem de jogo seja invencível;

REMOVEENEMIES: Remove todos os inimigos do jogo;

Implementação

SPAWNENEMIES: gera todo o tipo de inimigos, incluindo os inimigos de nível de dificuldade mais elevada;

SPAWNHEALTH: gera todo o tipo de *items* de vida (desde *items* que dão mais 10 de vida a *items* que dão mais 50 de vida e kits de vida que podem ser usados durante várias iterações);

SPAWNITEMS: gera vários tipos de *items* como o escudo, *jetpack*, tanque de ar, cartões de acesso, botas protetoras, hologramas, entre outros;

SPAWNWEAPONS: gera os vários tipos de armas e/ou munições;

Na lista apresentada abaixo estão representadas as especificações que ajudam a fazer as verificações das condições:

Player: contém toda a informação sobre a personagem de jogo, desde a vida, número de inimigos mortos, número de inimigos em volta, níveis de uso dos *items* até ao número de inimigos de nível de dificuldade mais aumentada que estão em volta;

ACTIONSTHREAD: permite consultar o número de *threads* que estão a ocorrer a cada minuto no jogo, por forma a não permitir que a mesma ação seja executada inúmeras vezes.

Sensor: contém a informação dos intervalos de valência e a atividade emocional do jogador.

4.4 Resumo

Neste capítulo foi descrita a implementação da metodologia no jogo EDuke32. Primeiro foi descrito o jogo usado como caso de estudo onde foi feita uma breve introdução ao mesmo. Depois foram apresentados os aspetos mais técnicos da implementação, tais como a arquitetura, o protocolo de comunicação e o sistema de regras. A arquitetura é composta pelas três componentes descritas na metodologia: jogo (Eduke32), *framework* e aplicação de obtenção de dados afetivos (Sensor Reader). Dentro da *framework*, foram ainda apresentadas as três componentes que a constituem e que permitem realizar as operações necessárias sobre as regras. A comunicação entre os componentes é feita através de *sockets*.

Implementação

Capítulo 5

Resultados

Neste capítulo será descrita numa primeira fase a forma como os testes experimentais foram realizados, seguindo-se dos resultados obtidos nos testes. No subcapítulo dos resultados, ainda será feita simultaneamente uma discussão que vai pender primeiro sobre os dados obtidos a partir dos formulários e em segundo lugar sobre os dados recolhidos pelos sensores ao longo das sessões.

5.1 *Setup Experimental*

Por forma a avaliar e validar a metodologia foram realizados vários testes. Para a realização dos testes foram utilizados vários sensores, nomeadamente: condutância da pele (SC¹), pressão sanguínea (BVP²) de onde foi extraído o ritmo cardíaco (HR³) e a EMG (eletromiografia) facial. Os sensores SC e BVP foram utilizados para medir a atividade das emoções do jogador, enquanto que a EMG facial foi utilizada com o propósito de calcular a valência.

O conjunto de sensores usado pertence ao equipamento neXus10 e o número de amostras por segundo (SPS⁴) foi o predefinido, respetivamente 2048 SPS para o EMG, 32 SPS para o GSR e 128 SPS para o BVP.

Por forma a calcular a valência e a atividade, foi usada a operação de normalização identificada na equação 5.1, cujos intervalos variavam entre 0 e 1, sendo que o intervalo [0.0,0.5] corresponde a valência e a atividade negativas e o intervalo [0.5,1.0] a valência e atividade positivas.

$$\frac{ValorAtual - ValorMínimo}{ValorMáximo - ValorMínimo} \quad (5.1)$$

O cálculo do valor de valência tem algumas particularidades que devem ser tomadas em atenção. Com vista a demonstrar essas particularidades, o algoritmo para transformar o sinal de EMG

¹Do inglês: Skin Conductance

²Do inglês: Blood Volume Pulse

³Do inglês: Heart Rate

⁴Do inglês: Samples per second

Resultados

em valência é apresentado em pseudo-código no algoritmo 1. Para o valor de atividade é apenas necessário ter em atenção o valor de HR visto que o sensor não era o mais adequado, pelo que para mudanças superiores a 20% em relação ao valor anterior foi usada a fórmula da equação 5.2 para corrigir eventuais mudanças bruscas.

$$HR = HR_ATUAL * 0.2 + HR_ANTERIOR * 0.8 \quad (5.2)$$

Algoritmo 1 Cálculo da valência com base na amplitude do sinal EMG

```
1: EMG_NEUTRO = 5
2: //Limitar valores de EMG1
3: SE EMG1 > MAX_EMG1 ENTÃO EMG1 = MAX_EMG1;
4: SE EMG1 < MIN_EMG1 ENTÃO EMG1 = MIN_EMG1;
5: //Limitar valores de EMG2
6: SE EMG2 > MAX_EMG2 ENTÃO EMG2 = MAX_EMG2;
7: SE EMG2 < MIN_EMG2 ENTÃO EMG2 = MIN_EMG2;
8: //Normalizar valores de acordo com os valores obtidos na calibração
9: EMG1 = NORMALIZAÇÃO(EMG1) //usando equação 5.1
10: EMG2 = NORMALIZAÇÃO(EMG2) //usando equação 5.1
11: //Não considerar valores de EMGX (X = 1 ou x=2) caso o seu valor seja demasiado baixo
12: SE EMG1 <= 0.01 ENTÃO EMG1 = 0
13: SE EMG2 <= 0.01 ENTÃO EMG2 = 0
14: //Se valor de EMGX for nulo então EMG = outro EMG
15: SE EMG1 == 0 ENTÃO EMG = EMG2
16: SE EMG2 == 0 ENTÃO EMG = EMG1
17: //Ajuste de intervalo de [0,1] para [0,10]
18: EMG1 = EMG_NEUTRO + (EMG_NEUTRO*EMG1)
19: EMG2 = EMG_NEUTRO - (EMG_NEUTRO*EMG2)
20: VALENCIA = (EMG1+EMG2)/2.0
21: SE EMG1 IGUAL A EMG_NEUTRO ENTÃO VALENCIA = EMG2
22: SE EMG2 IGUAL A EMG_NEUTRO ENTÃO VALENCIA = EMG1
23: //Normalizar o intervalo para que a valência varie de 0 a 1
24: VALENCIA = NORMALIZAÇÃO(VALENCIA) //usando equação 5.1
```

5.2 Validação e Avaliação

Para validar e avaliar os resultados foram recrutados 13 utilizadores que realizaram uma sessão de jogo no caso de estudo que foi apresentado na secção 4.1. Dos 13 indivíduos 10 eram do sexo masculino e os restantes eram do sexo feminino. Os indivíduos tinham idades compreendidas entre os 21 e 25 anos de idade. De entre todos os utilizadores é sabido que 6 deles (todos do sexo masculino) já tinham experiência neste jogo ou pelo menos em jogos deste género.

Cada sessão de jogo seguiu um conjunto de etapas. A figura 5.1 apresenta o protocolo experimental de cada uma destas sessões.

Resultados

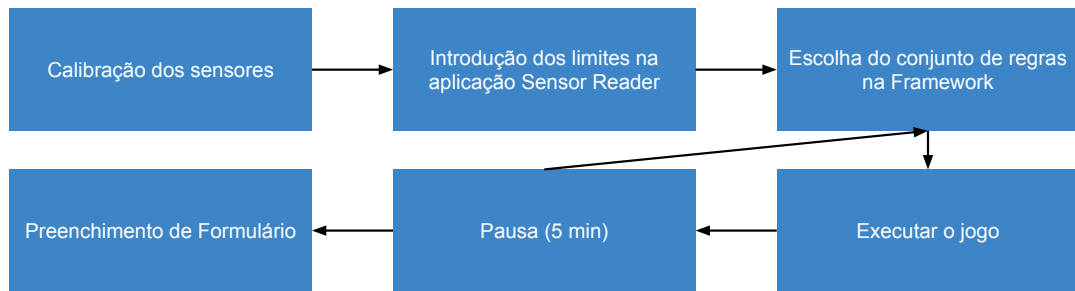


Figura 5.1: Protocolo Experimental

Na fase de calibração dos sensores, foi usada inicialmente uma música instrumental com o objetivo de acalmar o utilizador e obter o valor mínimo de ritmo cardíaco (HR) e de condutividade da pele (SC). De seguida foi mostrada uma apresentação em *powerpoint* que continha no primeiro *slide* uma figura do "Onde está o Wally" e uma vez mais uma música relaxante. Ao acabar a música do primeiro *slide* a apresentação prosseguia para um segundo *slide* onde é apresentada uma imagem e é reproduzido um som (grito), em que o objetivo era de aumentar os níveis de HR e SC. Para finalizar esta fase eram depois mostrados três vídeos, um de horror, outro de comédia e outro de drama. Após a reprodução dos vídeos, músicas e imagens foi efetuada uma análise visual dos valores captados pelos sensores durante a fase de calibração, sendo depois introduzidas estimativas para os valores máximos e mínimos dos sinais dos sensores na aplicação Sensor Reader.

Após estas duas primeiras fases foi iniciado um ciclo de jogo. O ciclo de jogo tinha quatro iterações, uma por cada conjunto de regras: Normal, Caos, Stress, Boredom. As sessões de jogo foram planeadas com antecedência para que a escolha dos conjuntos de regras fosse aleatória e não tivesse ocorrido em nenhuma outra sessão. Cada iteração era composta por várias etapas, sendo necessário primeiro escolher o conjunto de regras na framework, seguidamente escolher o conjunto de regras na aplicação Sensor reader, depois iniciar a recolha de dados dos sensores através da aplicação Biotrace e finalmente dar início ao jogo. Os conjuntos de regras jogados tinham várias diferenças, pelo que são descritas abaixo as regras contidas para cada um dos casos.

Normal: não foram feitas alterações no jogo, servindo este conjunto de regras principalmente para comparar o jogo original com o jogo modificado.

Boredom: de modo a tentar fazer com que a sessão fosse o mais aborrecida possível, foram retirados todos os inimigos do mapa, dadas todas as armas e foi ainda ativado um modo que impossibilita o jogador de morrer;

Caos: por forma a criar um situação caótica a iluminação era alterada entre mínimo e máximo de acordo com a valência - se a valência for menor ou igual a 0,4 (emoções negativas) a iluminação era máxima e era gerado um inimigo de 20 em 20 segundos com uma repetição de 2 vezes, enquanto que se fosse maior que 0,4 (emoções positivas) a iluminação passava a ser mínima. Para além das alterações de iluminação a valência foi usada também como

Resultados

condição de atribuição de armas ao jogador caso esta se situasse abaixo dos 0,6. Ainda a nível da iluminação, caso não fosse possível obter os valores de valência, ocorria uma alteração brusca e sucessiva entre iluminação mínima e máxima.

Por forma a que o jogador se mantenha em jogo durante mais tempo, quando o nível de vida está a um valor inferior a 50 era gerado um item que aumenta a vida do jogador em 50. O intervalo de vida pode variar entre 0 e 100 ou 0 e 200 ao possuir um item específico.

No final do nível para que seja mais complicado o jogador conseguir vencer, é ainda gerado um inimigo cujo nível de dificuldade é superior ao dos restantes, sendo ainda facultadas todas as armas, armadura e balas;

Stress: o nível de iluminação foi sempre mantido ao mínimo e no caso de o jogador ter um valor de vida superior a 90, eram gerados 2 inimigos de 20 em 20 segundos durante 10 iterações, sendo a dificuldade destes inimigos relativamente baixa.

Tal como no conjunto de regras Caos no final do nível também é gerado um inimigo com nível de dificuldade máximo, assim como as armas, balas e armadura suficientes para que o jogador consiga vencer o inimigo.

Uma vez mais é também gerado um item de vida sempre que o nível de vida do jogador se encontra num valor abaixo dos 50.

No início do nível o jogador tem acesso a uma arma com características superiores à arma com que é fornecida ao jogador inicialmente e que é suficiente para eliminar os inimigos gerados, pelo que eram ainda geradas munições para essa arma sempre que o número de munições estava abaixo de 10.

Finalmente por forma a causar uma situação que provoca ainda mais stress, se a atividade do jogador fosse relativamente baixa e a valência elevada e o jogador se encontrasse à frente da zona de fim de jogo, eram gerados mais inimigos (de nível relativamente fácil) para além do de nível mais difícil.

Uma vez que nem todos os jogadores tinham o mesmo à vontade para com o jogo, algumas das regras acima descritas tiveram algumas pequenas alterações em certas sessões. A maior parte destas pequenas alterações passaram pela redução de inimigos gerados por iteração, pelo *delay* e intervalo de tempo de geração de inimigos e pela geração de alguns *items* extra.

A última fase consistia no preenchimento de dois questionários. O primeiro pedia aos utilizadores para identificarem qual a sessão de jogo que tinham jogado em cada uma das quatro iterações. Nesta fase, os jogadores não sabiam ainda quais eram os conjuntos de regras que existiam. Já o segundo foi usado para obter informações e opiniões sobre a sessão de jogo.

Por forma a validar e avaliar os resultados foi feito um tratamento e análise a todos os dados recolhidos. Esta análise é apresentada no subcapítulo seguinte.

5.3 Análise dos resultados e Discussão

Neste subcapítulo segue-se uma apresentação dos resultados e uma discussão que apresenta argumentos a favor e razões pelas quais os resultados podem ou não ter sido os esperados.

Após terminarem a sessão os utilizadores que testaram a aplicação preencheram dois questionários. O primeiro questionário teve como objetivo identificar qual das quatro partes de jogo correspondiam a que conjunto de regras. Os resultados estão ilustrados na tabela 5.1.

	Normal	Caos	Stress	Boredom	% Identificação Correta
Utilizador 1	Normal	Stress	Caos	Boredom	50%
Utilizador 2	Normal	Caos	Stress	Boredom	100%
Utilizador 3	Normal	Caos	Stress	Boredom	100%
Utilizador 4	Caos	Normal	Stress	Boredom	50%
Utilizador 5	Normal	Stress	Caos	Boredom	50%
Utilizador 6	Normal	Stress	Caos	Boredom	50%
Utilizador 7	Normal	Stress	Caos	Boredom	50%
Utilizador 8	Normal	Caos	Stress	Boredom	100%
Utilizador 9	Normal	Caos	Stress	Boredom	100%
Utilizador 10	Normal	Caos	Stress	Boredom	100%
Utilizador 11	Normal	Caos	Stress	Boredom	100%
Utilizador 12	Normal	Caos	Stress	Boredom	100%
Utilizador 13	Normal	Caos	Stress	Boredom	100%

Tabela 5.1: Identificação de cada uma das partes da sessão

A verde estão as correspondências corretas, enquanto que a vermelho estão as correspondências erradas.

No caso dos utilizadores 1, 5, 6 e 7, o erro passou pela troca entre o conjunto de regras Caos e Stress. Este erro deveu-se provavelmente à semelhança entre as regras que constituíam cada um dos conjuntos. Por outro lado, para o caso do utilizador 4, o erro passou pela troca entre o conjunto de regras Caos e Normal. Neste caso, segundo o utilizador o erro deveu-se ao facto de o mesmo se sentir mais confortável ao jogar quando estava a ser usado o conjunto de regras Caos. Quanto aos restantes utilizadores, a identificação foi feita corretamente para todos os conjuntos de regras.

O segundo questionário era mais extenso e tinha como principal objetivo a avaliação de cada uma das partes da sessão pelos utilizadores. Em primeiro lugar os utilizadores foram questionados acerca de qual a parte que mais gostaram. Os resultados são apresentados na figura 5.2, onde a 1^a preferência representa o conjunto de regras que os utilizadores mais gostaram e a 4^a o que menos gostaram.

Resultados

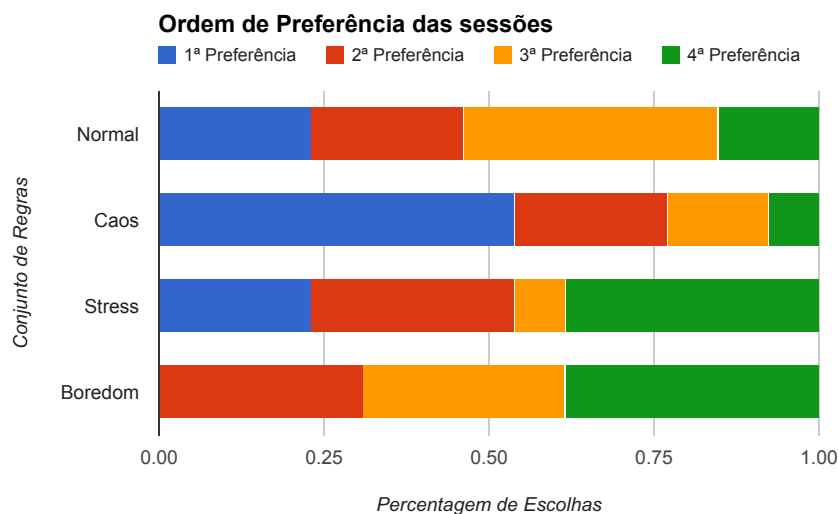


Figura 5.2: Ordem de preferência por partes de sessão

Como se pode observar o conjunto de regras preferido pela maioria dos utilizadores (aproximadamente 54%) foi o Caos. Já como segunda preferência (com cerca de 31%) houve um empate entre o Stress e o Boredom. Para a terceira preferência o conjunto preferido foi o Normal com aproximadamente 38%. Finalmente e como conjunto menos preferido, existe uma vez mais um empate entre Stress e Boredom.

Após questionados pela ordem de preferência entre os quatro conjuntos de regras, os utilizadores foram questionados sobre quais as emoções que tinham sentido ao longo de cada uma das partes. Seguem-se na tabela 5.2 os resultados desta questão.

	Normal	Caos	Stress	Boredom
Relaxado	61.54%	0.00%	23.08%	92.31%
Sobre Pressão	38.46%	69.23%	46.15%	0.00%
Nervoso	30.77%	76.92%	53.85%	0.00%
Ansioso	46.15%	61.54%	46.15%	23.08%
Tenso	15.38%	46.15%	46.15%	0.00%

Tabela 5.2: Percentagem de keywords por conjunto de regras

Nesta pergunta foi pedido aos utilizadores que escolhessem entre as emoções relaxado, sobre pressão, nervoso, ansioso e tenso, as que tinham sentido em cada uma das fases da sessão.

No contexto desta experiência, estar relaxado refere-se a uma sensação de bem estar em que o utilizador não tenha preocupações com o que possa ou não surgir, não tendo que prestar demasiada atenção ao que se está a passar no jogo.

Resultados

Estar sobre pressão refere-se a um estado em que o utilizador é atacado durante um longo período de tempo, obrigando-o a ter uma maior atenção ao que está a passar no jogo. O nervosismo assim como a pressão também se refere a um estado de alerta constante, mas neste caso é mais pelos inimigos que poderão ou não aparecer, sendo ainda possível também haverem outros fatores que contribuam para esta emoção, como por exemplo a personagem de jogo ter um nível de vida baixo enquanto está a ser atacado. Já a ansiedade está mais orientada para situações em que o jogador comece a estranhar a ausência de ação no jogo, como por exemplo a ausência de inimigos durante um período de tempo elevado. A tensão por outro lado refere-se a uma emoção em que o jogador está numa fase de concentração e alerta máximo.

Analisando a tabela (5.2) é possível verificar que para o conjunto de regras Boredom os utilizadores sentiram-se essencialmente relaxados, no entanto verifica-se também que muitos dos utilizadores se sentiram ansiosos. A principal razão para este facto deve-se a um bug do jogo, que obriga a que os inimigos apareçam visualmente antes de poderem ser removidos, o que normalmente fazia com que os jogadores disparassem para o inimigo até ao momento em que este desaparecia.

No caso do stress existiu uma grande diversificação de emoções. Embora o relaxamento não fosse algo previsto para este conjunto de regras, é sabido que os utilizadores que escolheram essa emoção tinham um maior à vontade com o jogo, por serem já terem alguma experiência em jogos do género.

Para o caos pelo contrário do stress não houve nenhum jogador que se sentisse relaxado, isto deve-se muito provavelmente ao facto da alteração nas condições de iluminação repentinas (que não aconteciam no stress), o que obrigou os jogadores a estarem mais atentos ao que se estava a passar.

Por outro lado na parte da sessão em que as condições de jogo eram as originais (sem alterações feitas pela *framework*), como era esperado, os jogadores tiveram uma vez mais uma grande diversificação de emoções, sendo a tensão a menos presente.

Ainda neste mesmo questionário foi pedido aos jogadores que classificassem o seu empenho e desempenho ao longo da sessão completa. Os resultados são apresentados na figura 5.3.

Para o desempenho foi usada uma escala de 1 a 5, onde 1 equivale a muito baixo, 2 a baixo, 3 a médio, 4 a bom e 5 a muito bom. A maior parte dos jogadores classificou o seu desempenho apenas como médio. Esta escolha deve-se provavelmente ao facto de em geral nos conjuntos de regras stress e caos os jogadores não se terem conseguido manter vivos durante um período prolongado.

Resultados

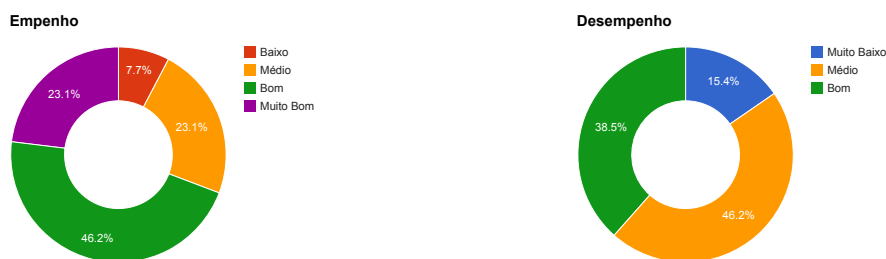


Figura 5.3: Empenho e Desempenho

Para o empenho foi usada a mesma escala que no desempenho. Em média o número de jogadores que achou que o seu empenho foi superior ou igual ao seu desempenho (cerca de 85%), foi muito superior ao oposto (15%, que acharam que o desempenho foi melhor que o empenho). No caso em que os jogadores consideraram ter obtido um desempenho superior ao empenho, os jogadores tinham mais à vontade com o jogo que nos restantes casos.

Uma outra questão foi quanto ao grau de divertimento de cada sessão, em que foi pedido aos jogadores para indicarem para cada uma das partes da sessão se tinha sido Muito Divertida, Divertida ou Aborrecida. Para avaliar as respostas, foi usado um teste de *Friedman* [Gal10] por forma a verificar se os utilizadores têm níveis de preferência suficientemente diferentes entre os conjuntos de regras.

Pares de Conjuntos de Regras	Valores de P
Caos - Boredom	0.03391221
Normal - Boredom	0.02055857
Stress - Boredom	0.55317206
Normal - Caos	0.99842018
Stress - Caos	0.50093788
Stress - Normal	0.40063575

Tabela 5.3: Resultados do teste de Friedman

Através dos resultados obtidos pelos testes de *Friedman* (tabela 5.3) é possível concluir que existe uma diferença significativa entre alguns dos quatro conjuntos de regras, $\chi^2 = 10.89$, $p = 0.0123$. A análise *posthoc* mostra que esta diferença se nota principalmente entre os conjuntos de regras Normal-Boredom (valor de P aproximadamente 0.0206) e no par Caos - Boredom (valor de P aproximadamente 0.034).

Neste segundo questionário após uma breve explicação da metodologia e da solução desenvolvida, foi pedido aos utilizadores que respondessem a uma questão de resposta aberta indicando

Resultados

se achavam útil e o porquê. Em todos os casos a resposta foi positiva e teve como principal argumento que ferramentas deste género podem ajudar os *designers* a criar jogos mais adaptados aos jogadores e às suas emoções.

Concluída a interpretação dos resultados obtidos através dos dados dos questionários, segue-se agora uma interpretação dos valores obtidos através dos sensores ao longo de cada uma das sessões.

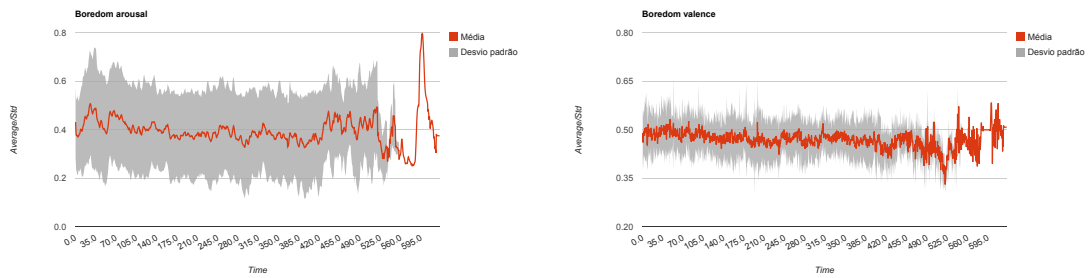


Figura 5.4: Atividade e valência para conjunto de regras Boredom

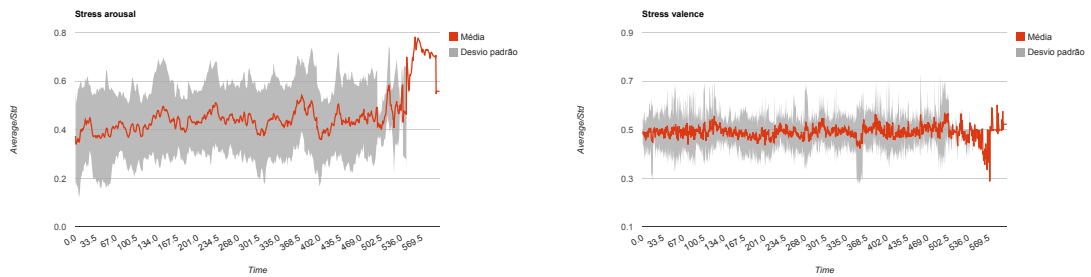


Figura 5.5: Atividade e valência para conjunto de regras Stress

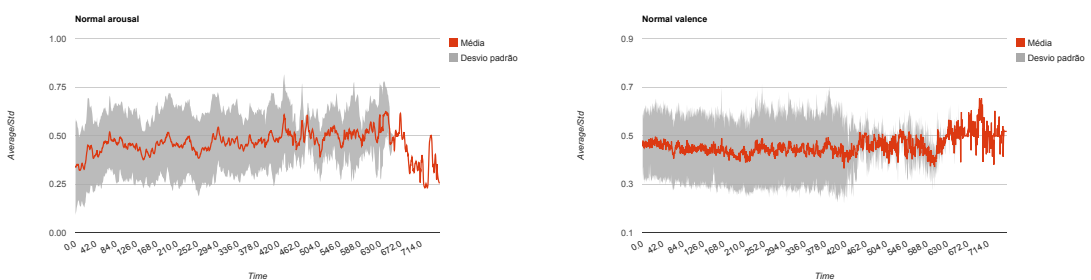


Figura 5.6: Atividade e valência para conjunto de regras Normal

Nos gráficos 5.4, 5.5, 5.6, 5.7 estão representados os dados recolhidos através dos sensores. As linhas a vermelho representam os valores médios de valência e atividade de todos os utilizadores para cada um dos conjuntos de regras, enquanto que a cinzento estão representados os desvios padrão.

Resultados

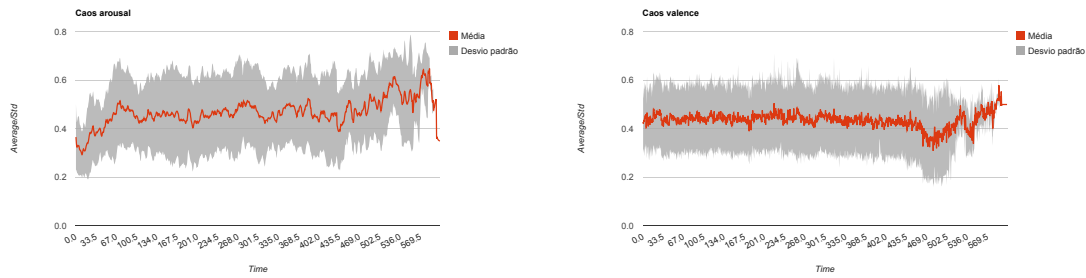


Figura 5.7: Atividade e valência para conjunto de regras Caos

Através da observação destes gráficos é possível concluir que não existem diferenças significativas entre os valores médios para os diferentes conjuntos de regras. No entanto, é também possível observar que o desvio padrão tem valores bastante elevados. As razões apontadas para estes resultados são possíveis falhas no processo de calibração dos sensores e o mau funcionamento de alguns dos sensores durante os testes. Para uma melhor compreensão destes problemas foram introduzidos nos anexos os gráficos A.1 a A.13 que representam os dados recolhidos por indivíduo.

Estas falhas relativas aos dados recolhidos pelos sensores não afetaram no entanto os resultados obtidos através dos questionários, podendo no entanto ter afetado a frequência com que certas regras eram verificadas.

5.4 Conclusões

Após a interpretação dos resultados obtidos através dos testes e dos dados recolhidos pelos sensores, é possível concluir a validação não foi propriamente concluída com sucesso, uma vez que era esperado existirem diferenças muito mais significativas para os dados captados pelos sensores entre os conjuntos de regras escolhidos. É no entanto importante mencionar que estes resultados pouco satisfatórios se deveram acima de tudo à forma como os sensores foram calibrados e a falhas nos mesmos ao longo dos testes.

A avaliação que foi feita a partir dos questionários, pelo contrário dos resultados da validação obteve resultados muito mais satisfatórios. Como previsto quando foram escolhidos os conjuntos de regras, os indivíduos que testaram a aplicação conseguiram notar diferenças significativas entre cada uma das partes das sessões.

Outro ponto importante foi o facto de a maior parte dos utilizadores (cerca de 62%) conseguirem identificar corretamente o conjunto de regras escolhido. Estes resultados são especialmente importantes porque a escolha das regras não foi efetuada por alguém da área de *design*, o que muito provavelmente faria aumentar a percentagem de sucesso na identificação dos conjuntos de regras.

Capítulo 6

Conclusões e Trabalho Futuro

O trabalho desenvolvido nesta dissertação consistiu no desenvolvimento e implementação de uma metodologia que tem como principal objetivo criar ferramentas que apoiem o *design* de videogames. Esta metodologia permite aos *designers* se focarem nos estados emocionais que pretendem transmitir aos jogadores.

Neste documento, primeiro foi efetuado um levantamento do estado da arte do reconhecimento de emoções, geração procedimental e ainda da geração procedimental baseada em dados afetivos. Seguidamente foi descrita a metodologia, começando pela arquitetura e partindo depois para cada uma das componentes, uma breve descrição dos possíveis tipos de regras e finalmente alguns casos de aplicação. Depois foi descrita a implementação do caso de estudo que foi usado para testar a metodologia. Finalmente foram analisados os resultados obtidos com os testes feitos à implementação da metodologia.

6.1 Satisfação dos Objetivos

Com base nos resultados obtidos a partir dos questionários preenchidos pelos utilizadores que testaram a aplicação, é possível concluir que a metodologia foi implementada e definida com sucesso. Os resultados mostram que é possível usar um sistema de regras por forma a permitir uma rápida prototipagem e alteração do conteúdo gerado num jogo e que os jogadores conseguem distinguir e aperceber-se das mudanças efetuadas por essas regras.

Por outro lado, com os resultados obtidos a partir dos sensores, não foi possível obter resultados conclusivos devido aos problemas que ocorreram durante os testes.

Ao implementar esta metodologia no jogo Eduke32 e com base nos exemplos indicados na secção 3.4 foi mostrado ainda que esta metodologia é suficientemente genérica para aplicar a um grande número de jogos. Dado ainda o tempo em que decorreu a implementação, é também possível concluir que o esforço e tempo necessários à implementação não deverão afetar os custos de desenvolvimento do jogo.

6.2 Trabalho Futuro

Apesar de algumas das características já demonstradas ao longo deste trabalho, as potencialidades desta metodologia são muito maiores, limitando-se sobretudo à imaginação dos designers e dos programadores de jogos.

Durante a implementação e o levantamento do estado da arte foram encontradas algumas dificuldades que se tornaram em oportunidades de trabalho futuro. Seguem-se portanto algumas propostas de tarefas a realizar por forma a continuar este trabalho.

Em primeiro lugar de modo a evitar o problema de os jogadores poderem repetir uma certa estratégia para progredir com sucesso no jogo respetivamente [LB11b], seria interessante obter modelos de estratégias usadas pelos jogadores e através desses modelos alterar a geração de conteúdo.

Tal como aconteceu em alguns testes por vezes os jogadores não conseguem atingir determinados objetivos, pelo que seria interessante que existissem regras que adaptassem o jogo por forma a permitir que os objetivos fossem alcançados ou até mesmo de forma a que os objetivos se adaptassem às dificuldades dos jogadores.

Para melhorar a compreensão do estado de jogo poderá ser adicionada alguma semântica que permita identificar mais facilmente espaços e outros componentes do jogo. Um exemplo de uso desta funcionalidade seria saber se jogador está num espaço grande ou pequeno o que permitiria gerar conteúdo com base nessa informação.

Uma outra funcionalidade que seria uma mais valia para os jogos atuais, seria usar o sistema de regras para alterar a história do jogo ou criar novas missões.

No caso da metodologia poderá ser implementado um sistema de alteração de regras dinâmico que deverá alterar as regras automaticamente conforme for necessário, baseando-se por exemplo no comportamento e ações do jogador.

Outra funcionalidade seria a especificação de regras baseado no nível de dificuldade, isto é, permitir a criação de regras diferentes consoante o nível de dificuldade do jogo.

Um dos problemas que surgiram deveu-se ao facto de a taxa de atualização do estado de jogo ser diferente da taxa de atualização dos dados afetivos. Com vista a resolver este problema, deverá ser implementada uma funcionalidade que permita ao designer controlar quando devem ser verificadas as regras. Um exemplo de um problema que ocorreu durante os testes foi na geração de um item que aumentava a vida do personagem de jogo quando este tinha esse nível abaixo de 50, em que a regra era classificada como verdadeira duas ou mais vezes gerando o item múltiplas vezes.

Com vista a melhorar a qualidade de avaliação das regras, seria ainda uma mais valia adicionar à *framework* um funcionalidade que permitisse identificar para cada intervalo de tempo e espaço do mundo virtual, quais as regras que foram verificadas e ainda o estado afetivo do jogador.

Por forma a facilitar ainda mais a implementação desta metodologia, seria ainda interessante construir uma nova versão da *framework* descrita neste trabalho com suporte a *plugins* e com

Conclusões e Trabalho Futuro

uma plataforma que permita aos designers partilharem bibliotecas de regras que sejam genéricas o suficiente para qualquer jogo que seja do mesmo género do desenvolvido pelos mesmos.

Finalmente, seria ainda uma mais valia realizar um estudo de como esta metodologia poderia ser utilizada em jogos *multiplayer* e como seriam resolvidos problemas como diferentes estados afetivos entre os diferentes jogadores e outros problemas inerentes deste modo de jogo.

Conclusões e Trabalho Futuro

Referências

- [AGB05] D.A. Ashlock, S.P. Gent e K.M. Bryden. Evolution of L-systems for Compact Virtual Landscape Generation. In *2005 IEEE Congress on Evolutionary Computation*, volume 3, pages 2760–2767. IEEE, 2005. URL: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1555041>, doi:10.1109/CEC.2005.1555041.
- [Amb11] Mike Ambinder. Biofeedback in gameplay: How valve measures physiology to enhance gaming experience. In *Game Developers Conference*, 2011.
- [BGP11] L Brown, B Grundlehner e J Penders. Towards wireless emotional valence detection from EEG. In *Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE*, pages 2188–2191, 2011. doi:10.1109/IEMBS.2011.6090412.
- [BW05] X. Baele e N. Warzée. Real time L-system generated trees based on modern graphics hardware. In *International Conference on Shape Modeling and Applications 2005 (SMI' 05)*, pages 184–193. IEEE Comput. Soc, 2005. URL: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1563224>, doi:10.1109/SMI.2005.38.
- [CBPD11] D M D Carli, F Bevilacqua, C T Pozzer e M C D’Ornellas. A Survey of Procedural Content Generation Techniques Suitable to Game Development. In *Games and Digital Entertainment (SBGAMES), 2011 Brazilian Symposium on*, pages 26–35, 2011. doi:10.1109/SBGAMES.2011.15.
- [CKGP06] Guillaume Chanel, Julien Kronegg, Didier Grandjean e Thierry Pun. Emotion assessment: Arousal evaluation using EEG’s and peripheral physiological signals. In *Multimedia content representation, classification and security*, pages 530–537. Springer, 2006.
- [CM09] Cristina Conati e Heather Maclaren. Modeling User Affect from Causes and Effects. In Geert-Jan Houben, Gord McCalla, Fabio Pianesi e Massimo Zancanaro, editors, *User Modeling, Adaptation, and Personalization*, volume 5535, chapter 4, pages 4–15. Springer Berlin Heidelberg, 2009. URL: http://dx.doi.org/10.1007/978-3-642-02247-0_4, doi:10.1007/978-3-642-02247-0_4.
- [DZZ10] Wensheng Dong, Xinyan Zhang e Caijun Zhang. Generation of Cloud Image Based on Perlin Noise. In *2010 International Conference on Multimedia Communications*, pages 61–63. IEEE, August 2010. URL: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5694143>, doi:10.1109/MEDIACOM.2010.77.

REFERÊNCIAS

- [ELM11] Clayton Epp, Michael Lippold e Regan L Mandryk. Identifying emotional states using keystroke dynamics, 2011. doi:10.1145/1978942.1979046.
- [Gal10] Tal Galili. POST HOC ANALYSIS FOR FRIEDMAN’S TEST (R CODE), 2010. URL: <http://www.r-statistics.com/2010/02/post-hoc-analysis-for-friedmans-test-r-code/>.
- [GDA05] Kiel Mark Gilleade, Alan Dix e Jen Allanson. Affective videogames and modes of affective gaming: assist me, challenge me, emote me. *DiGRA 2005: Changing Views – Worlds in Play.*, 2005.
- [Gol89] David E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1st edition, 1989.
- [Haz06] Richard L. Hazlett. Measuring emotional valence during interactive experiences: Boys at video game play. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI ’06, pages 1023–1026. ACM, New York, NY, USA, 2006. URL: <http://doi.acm.org/10.1145/1124772.1124925>, doi:10.1145/1124772.1124925.
- [Hen12] Hentzau. The Procedural Generation. *scientificgamer*, 2012. URL: <http://scientificgamer.com/the-procedural-generation/>.
- [HGSW04] Andreas Haag, Silke Goronzy, Peter Schaich e Jason Williams. Emotion Recognition Using Bio-sensors: First Steps towards an Automatic System. In Elisabeth André, Laila Dybkjær, Wolfgang Minker e Paul Heisterkamp, editors, *Affective Dialogue Systems*, volume 3068, chapter 4, pages 36–48. Springer Berlin Heidelberg, 2004. URL: http://dx.doi.org/10.1007/978-3-540-24842-2_4, doi:10.1007/978-3-540-24842-2_4.
- [HKA05] M Mahmud Hasan, M Sazzad Karim e Emdad Ahmed. Generating and Rendering Procedural Clouds in Real Time on Programmable 3D Graphics Hardware. In *2005 Pakistan Section Multitopic Conference*, pages 1–6. IEEE, December 2005. URL: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4133457>, doi:10.1109/INMIC.2005.334442.
- [Hud08] Eva Hudlicka. Affective computing for game design. In *Proceedings of the 4th Intl. North American Conference on Intelligent Games and Simulation (GAMEON-NA), Montreal, Canada*, 2008.
- [JW03] Daniel M. Johnson e Janet Wiles. Effective affective user interface design in games. *Ergonomics*, 46(13-14):1332–1345, 2003. doi:10.1080/00140130310001610865.
- [Kar11] Jarkko Kari. Cellular Automata: Tutorial. 2011. URL: <http://ebookbrowse.net/caintro-pdf-d53148954>.
- [KTTY12] Manuel Kerssemakers, Jeppe Tuxen, Julian Togelius e Georgios N. Yannakakis. A procedural procedural level generator generator. In *2012 IEEE Conference on Computational Intelligence and Games (CIG)*, pages 335–341. IEEE, September 2012. URL: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6374174>, doi:10.1109/CIG.2012.6374174.

REFERÊNCIAS

- [Lř2] Hugo Lřvheim. A new three-dimensional model for emotions and monoamine neurotransmitters. *Medical hypotheses*, 78(2):341–8, February 2012. URL: <http://www.sciencedirect.com/science/article/pii/S0306987711005883>, doi:10.1016/j.mehy.2011.11.016.
- [LB11a] Ricardo Lopes e Rafael Bidarra. A semantic generation framework for enabling adaptive game worlds. In *Proceedings of the 8th International Conference on Advances in Computer Entertainment Technology - ACE '11*, page 1, New York, New York, USA, November 2011. ACM Press. URL: <http://dl.acm.org/citation.cfm?id=2071423.2071431>, doi:10.1145/2071423.2071431.
- [LB11b] Ricardo Lopes e Rafael Bidarra. Adaptivity Challenges in Games and Simulations: A Survey. *IEEE Transactions on Computational Intelligence and AI in Games*, 3(2):85–99, June 2011. URL: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5765665>, doi:10.1109/TCIAIG.2011.2152841.
- [LCCS07] Enrique Leon, Graham Clarke, Victor Callaghan e Francisco Sepulveda. A user-independent real-time emotion recognition system for software agents in domestic environments. *Engineering Applications of Artificial Intelligence*, 20(3):337–345, 2007. URL: <http://www.sciencedirect.com/science/article/pii/S0952197606001011>, doi:http://dx.doi.org/10.1016/j.engappai.2006.06.001.
- [LTB12] Ricardo Lopes, Tim Tutenel e Rafael Bidarra. Using gameplay semantics to procedurally generate player-matching game worlds. In *Proceedings of PCG 2012 - Workshop on Procedural Content Generation for Games, co-located with the Seventh International Conference on the Foundations of Digital Games*, Raleigh, NC, 2012. URL: <http://graphics.tudelft.nl/Publications-new/2012/LTB12a>.
- [MA07] Regan L Mandryk e M Stella Atkins. A fuzzy physiological approach for continuously modeling emotion during interaction with play technologies. *International Journal of Human-Computer Studies*, 65(4):329–347, 2007. URL: <http://www.sciencedirect.com/science/article/pii/S1071581906001881>, doi:http://dx.doi.org/10.1016/j.ijhcs.2006.11.011.
- [MCSL09] Dan Meng, Xingquan Cai, Zhitong Su e Jinhong Li. Photorealistic terrain generation method based on fractal geometry theory and procedural texture. In *2009 2nd IEEE International Conference on Computer Science and Information Technology*, pages 341–344. IEEE, 2009. URL: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5234644>, doi:10.1109/ICCSIT.2009.5234644.
- [MRNY07] M Murugappan, M Rizon, R Nagarajan e S Yaacob. EEG feature extraction for classifying emotions using FCM and FKM. *International Journal of Computers and Communications*, 1(2):21–25, 2007.
- [NARO14] Pedro Alves Nogueira, Růben Aguiar, Rui Rodrigues e Eugénio Oliveira. Computational Models of Players' Physiological-based Emotional Reactions: A Digital Games Case Study. In *Proceedings of the 2014 IEEE/WIC/ACM International Conference on Intelligent Agent Technology*, 2014.

REFERÊNCIAS

- [NLAF03] Fatma Nasoz, Christine L Lisetti, Kaye Alvarez e Neal Finkelstein. Emotion recognition from physiological signals for user modeling of affect. In *Proceedings of the 3rd Workshop on Affective and Attitude User Modelling (Pittsburgh, PA, USA, 2003)*.
- [NRON13a] Pedro Alves Nogueira, Rui Rodrigues, Eugénio Oliveira e Lennart Nacke. Guided Emotional State Regulation: Understanding and Shaping Players' Affective Experiences in Digital Games, 2013. URL: <http://www.aaai.org/ocs/index.php/AIIDE/AIIDE13/paper/view/7367>.
- [NRON13b] Pedro Alves Nogueira, Rui Rodrigues, Eugénio Oliveira e Lennart E. Nacke. A Hybrid Approach at Emotional State Detection: Merging Theoretical Models of Emotion with Data-Driven Statistical Classifiers. In *Web Intelligence (WI) and Intelligent Agent Technologies (IAT), 2013 IEEE/WIC/ACM International Joint Conferences on*, volume 2, pages 253–260, November 2013. doi:10.1109/WI-IAT.2013.117.
- [NRON13c] Pedro Alves Nogueira, Rui Rodrigues, Eugénio Oliveira e Lennart E. Nacke. A Regression-based Method for Lightweight Emotional State Detection in Interactive Environments. In *Proceedings of the XVI Portuguese Conference on Artificial Intelligence (EPIA), Angra do Heroísmo, Açores, Portugal, 2013*.
- [OCC90] Andrew Ortony, Gerald L. Clore e Allan Collins. *The cognitive structure of emotions*. Cambridge university press, 1990.
- [PL90] P. Prusinkiewicz e Aristid Lindenmayer. *The Algorithmic Beauty of Plants*. Springer-Verlag New York, Inc., New York, NY, USA, 1990.
- [Plu91] Robert Plutchik. *The emotions*. University Press of America, 1991.
- [SMBC13] Pedro Silva, Pascal Müller, Rafael Bidarra e Antonio Coelho. Node-based shape grammar representation and editing. In *Proceedings of PCG 2013 - Workshop on Procedural Content Generation for Games, co-located with the Eighth International Conference on the Foundations of Digital Games*, Chania, Crete, Greece, may 2013. URL: <http://graphics.tudelft.nl/Publications-new/2013/SMBC13a>.
- [SN13] José Serra e Pedro Nogueira. Personality Simulation In Interactive Agents Through Emotional Bias. In *European Conference on Modelling and Simulation*, pages 25–31. IEEE, 2013.
- [ST01] Louis A Schmidt e Laurel J Trainor. Frontal brain electrical activity (EEG) distinguishes valence and intensity of musical emotions. *Cognition & Emotion*, 15(4):487–500, 2001. URL: <http://dx.doi.org/10.1080/02699930126048>, doi:10.1080/02699930126048.
- [TNS13] Luís Filipe Teófilo, Pedro Alves Nogueira e Pedro Brandão Silva. GEMINI: A Generic Multi-Modal Natural Interface Framework for Videogames. In *Advances in Information Systems and Technologies*, pages 873–884. Springer, 2013.
- [TSBdK09] Tim Tutenel, Ruben Michaël Smelik, Rafael Bidarra e Klaas Jan de Kraker. Using Semantics to Improve the Design of Game Worlds. In *AIIDE*, 2009.

REFERÊNCIAS

- [TSL⁺11] Tim Tutenel, Ruben M. Smelik, Ricardo Lopes, Klaas Jan de Kraker e Rafael Bidarra. Generating Consistent Buildings: A Semantic Approach for Integrating Procedural Techniques. *IEEE Transactions on Computational Intelligence and AI in Games*, 3(3):274–288, September 2011. URL: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5960781>, doi:10.1109/TCIAIG.2011.2162842.
- [TYSB11] J Togelius, G N Yannakakis, K O Stanley e C Browne. Search-Based Procedural Content Generation: A Taxonomy and Survey. *Computational Intelligence and AI in Games, IEEE Transactions on*, 3(3):172–186, 2011. doi:10.1109/TCIAIG.2011.2148116.
- [vdLLB14] Roland van der Linden, Ricardo Lopes e Rafael Bidarra. Procedural Generation of Dungeons. *IEEE Transactions on Computational Intelligence and AI in Games*, 6(1):78–89, March 2014. URL: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6661386>, doi:10.1109/TCIAIG.2013.2290371.
- [vLLB13] Roland van Linden, Ricardo Lopes e Rafael Bidarra. Designing procedurally generated levels. In *Proceedings of the the second workshop on Artificial Intelligence in the Game Design Process. 2013.*, 2013. URL: <http://graphics.tudelft.nl/Publications-new/2013/LLB13a>.
- [Wil] Daniel Williamson. Duke Nukem 3D (Video Game 1996) - IMDb. URL: <http://www.imdb.com/title/tt0228246/>.
- [WKW13] Shunli Wang, Fengju Kang e Dinghua Wang. Ocean wave real-time simulation based on adaptive fusion. In *Control Conference (CCC), 2013 32nd Chinese, IEEE, 2013.*, pages 8557–8560, 2013.
- [YT11] G. N. Yannakakis e Julian Togelius. Experience-Driven Procedural Content Generation. *Affective Computing, IEEE Transactions on*, 2(3):147–161, 2011. doi:10.1109/T-AFFC.2011.6.

REFERÊNCIAS

Anexo A

Anexos

Listagem A.1 Mensagem que representa uma parte do estado de jogo.

```
1: {
2:   {
3:     "REQUEST_TYPE": "1",
4:     "DATA": [
5:       {
6:         "OBJECTTYPE" : "GAME_OBJECTS",
7:         "OBJECT": "PLAYER",
8:         "ATTRIBUTES": [
9:           {"HEALTH": "100"},
10:          {"AMMO": "200"}
11:        ]
12:      },
13:      {
14:        "OBJECTTYPE" : "GAME_OBJECTS",
15:        "OBJECT": "BOSS",
16:        "ATTRIBUTES": [
17:          {"HEALTH": "100"},
18:          {"AMMO": "200"}
19:        ]
20:      }
21:    ]
22:  }
23: }
24: }
```

Anexos

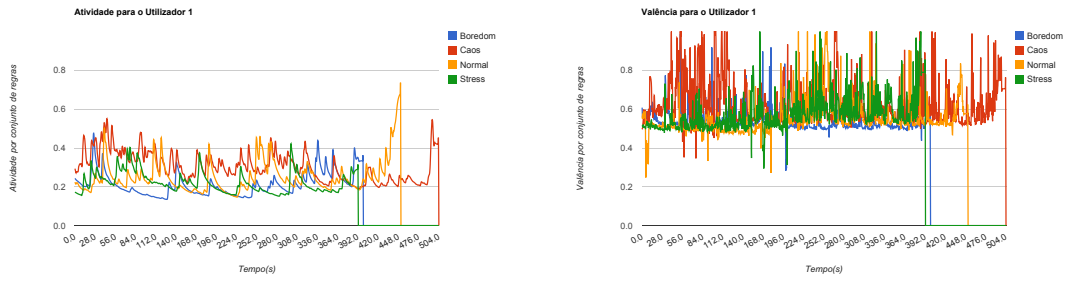


Figura A.1: Atividade e valência para o utilizador 1

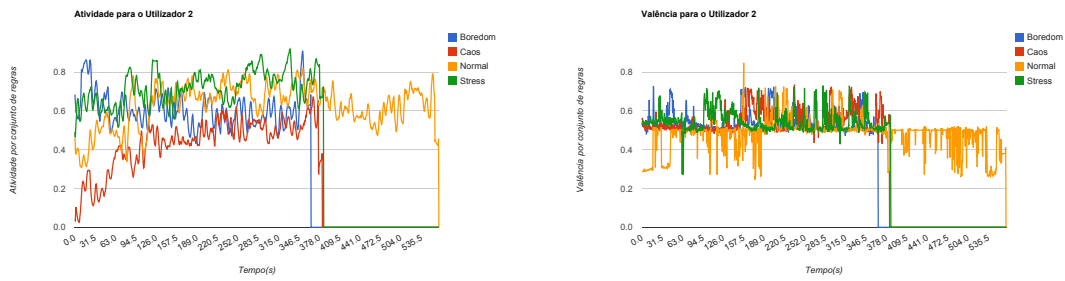


Figura A.2: Atividade e valência para o utilizador 2

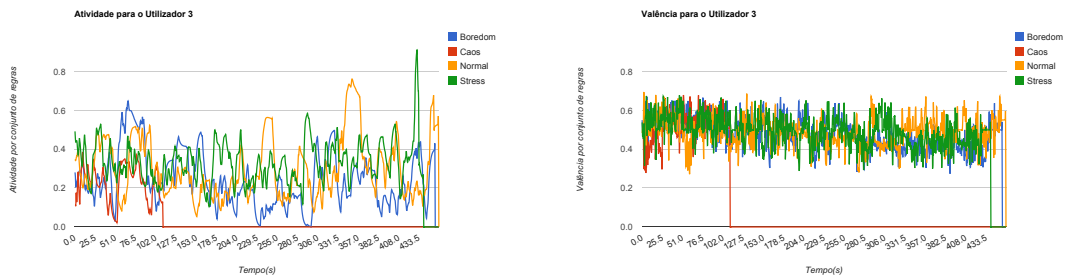


Figura A.3: Atividade e valência para o utilizador 3

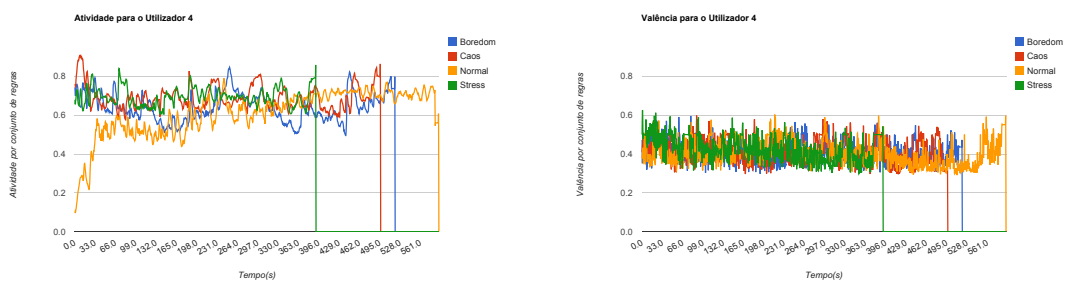


Figura A.4: Atividade e valência para o utilizador 4

Anexos

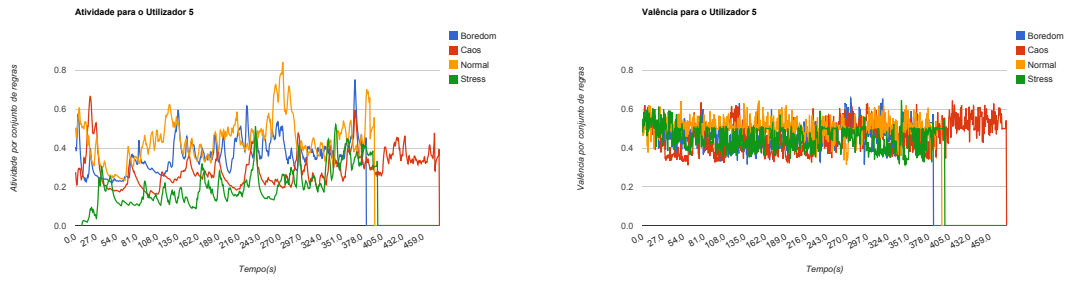


Figura A.5: Atividade e valência para o utilizador 5

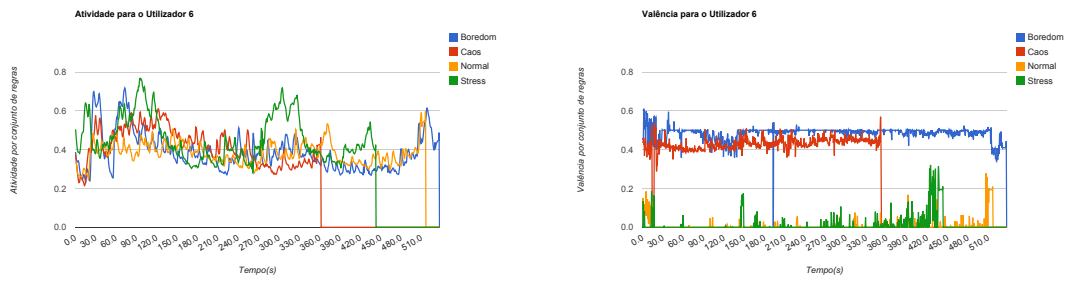


Figura A.6: Atividade e valência para o utilizador 6

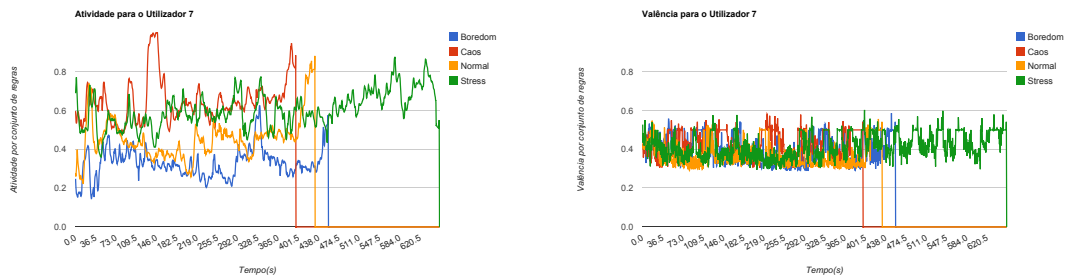


Figura A.7: Atividade e valência para o utilizador 7

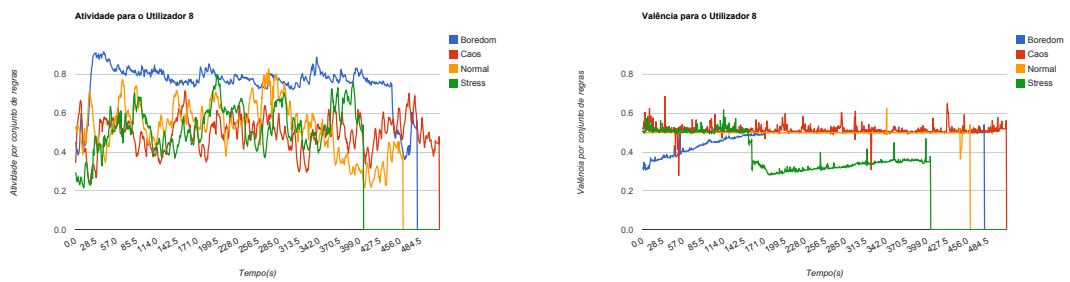


Figura A.8: Atividade e valência para o utilizador 8

Anexos

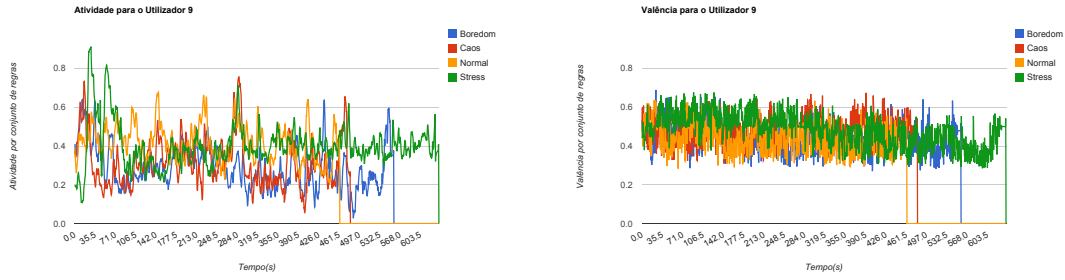


Figura A.9: Atividade e valência para o utilizador 9

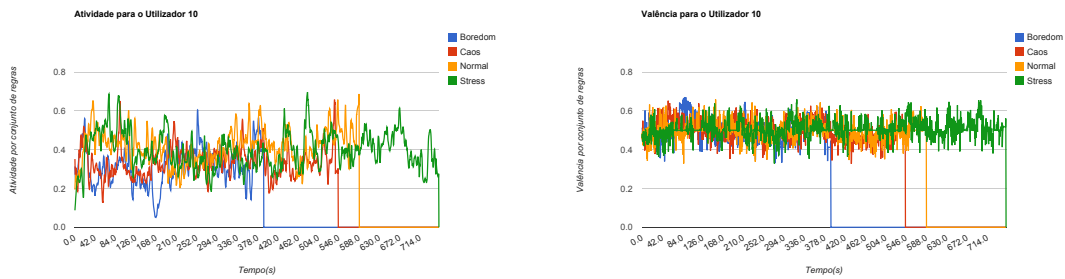


Figura A.10: Atividade e valência para o utilizador 10

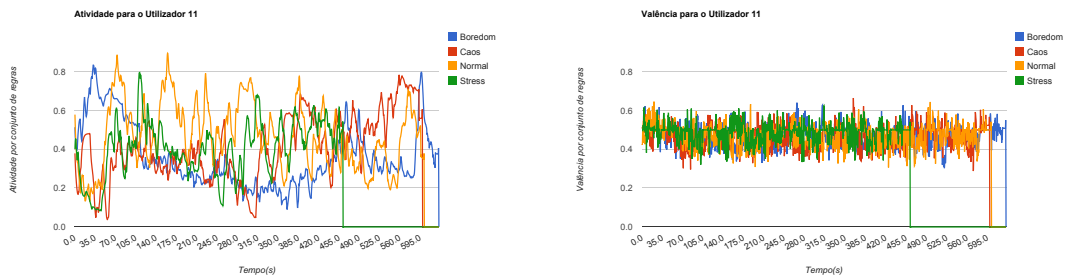


Figura A.11: Atividade e valência para o utilizador 11

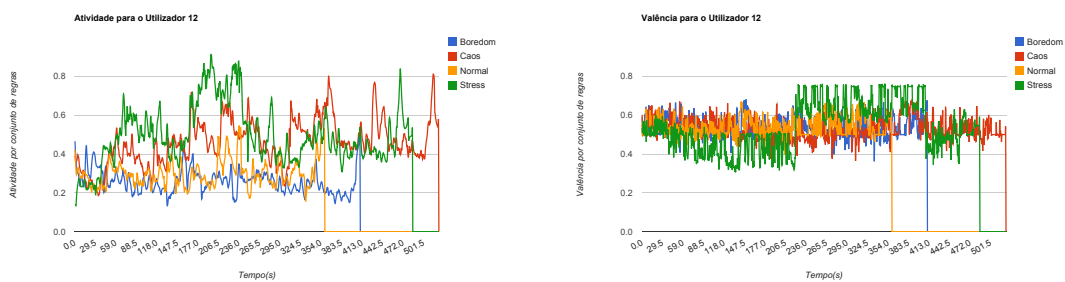


Figura A.12: Atividade e valência para o utilizador 12

Anexos

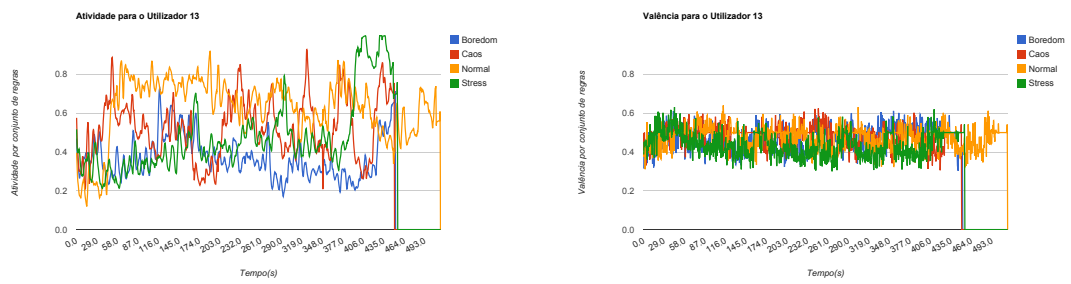


Figura A.13: Atividade e valência para o utilizador 13

Anexos

Índice Remissivo

A

ACE, [19](#)
Afetivo, [6](#), [20](#), [25–27](#), [29](#), [33](#), [37](#), [49](#), [50](#)
Atividade, [5](#), [6](#), [11](#), [12](#), [20](#), [39](#), [40](#), [42](#), [47](#)
AV, [5](#), [6](#)

C

Cubo de Lövhelm, [6](#), [23](#)

E

EDuke32, [31](#), [33](#), [37](#), [49](#)
EMG, [8](#), [11](#), [12](#), [20](#), [39](#)

G

GSR, [9–11](#), [39](#)

H

HR, [9](#), [11](#), [40](#)

O

OCC, [4](#), [11](#), [23](#)

P

Procedimental, [3](#), [12](#), [15](#), [16](#), [18–20](#), [23](#), [49](#)

R

Roda das emoções, [5](#)

V

Valência, [4–6](#), [11](#), [12](#), [39–42](#), [47](#)